



TUGAS AKHIR TF 141581

RANCANG BANGUN SISTEM PENGENDALI KADAR OKSIGEN TERLARUT DENGAN ALGORITMA *FUZZY LOGIC CONTROLLER* PADA BUDIDAYA AKUAPONIK

ALBERTO RIOLLY CAHYANTARA
NRP 2412 100 060

Dosen Pembimbing
Hendra Cordova, S.T., M.T.

DEPARTEMEN TEKNIK FISIKA
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR TF 141581

RANCANG BANGUN SISTEM PENGENDALI KADAR OKSIGEN TERLARUT DENGAN ALGORITMA *FUZZY LOGIC CONTROLLER* PADA BUDIDAYA AKUAPONIK

ALBERTO RIOLLY CAHYANTARA
NRP 2412 100 060

Dosen Pembimbing
Hendra Cordova, S.T., M.T.

DEPARTEMEN TEKNIK FISIKA
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT TF 141581

DESIGN OF DISSOLVED OXYGEN CONTROL SYSTEM USING FUZZY LOGIC CONTROLLER ALGORITHM ON AQUAPONIC

***ALBERTO RIOLLY CAHYANTARA
NRP 2412 100 060***

*Supervisor
Hendra Cordova, S.T., M.T.*

***ENGINEERING PHYSICS DEPARTMENT
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya 2017***

PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini:

Nama : Alberto Riolly Cahyantara
NRP : 2412100060
Departemen / Prodi : Teknik Fisika / S1 Teknik Fisika
Fakultas : Fakultas Teknologi Industri
Perguruan Tinggi : Institut Teknologi Sepuluh Nopember

Dengan ini menyatakan bahwa Tugas Akhir dengan judul “Rancang Bangun Sistem Pengendali Kadar Oksigen Terlarut dengan Algoritma *Fuzzy Logic Controller* pada Budidaya Akuaponik” adalah benar karya saya sendiri dan bukan plagiat dari karya orang lain. Apabila di kemudian hari terbukti terdapat plagiat pada Tugas Akhir ini, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenar-benarnya.

Surabaya, Juli 2017
Yang membuat pernyataan,



Alberto Riolly Cahyantara
NRP. 2412100060

LEMBAR PENGESAHAN

RANCANG BANGUN SISTEM PENGENDALI KADAR OKSIGEN TERLARUT DENGAN ALGORITMA FUZZY LOGIC CONTROLLER PADA BUDIDAYA AKUAPONIK

Oleh:

Alberto Riolly Cahvantara
NRP. 2412 100 060

Surabaya, 27 Juli 2017
Mengetahui/Menyetujui

Pembimbing



Hendra Cordova, S.T., MT.
NIP. 19690530 199412 1 001

Kepala Departemen
Teknik Informatika FTI-ITS



Agus Muhammad Hatta, S.T., M.Si. Ph.D
NIP. 19780902 200312 1 002

**RANCANG BANGUN SISTEM PENGENDALI KADAR
OKSIGEN TERLARUT DENGAN ALGORITMA
FUZZY LOGIC CONTROLLER PADA BUDIDAYA
AKUAPONIK**

TUGAS AKHIR

Diajukan untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
pada
Bidang Studi Rekayasa Instrumentasi dan Kontrol
Program Studi S-1 Departemen Teknik Fisika
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Oleh:

ALBERTO RIOLLY CAHYANTARA
NRP. 2412 100 060

Disetujui oleh Tim Penguji Tugas Akhir:

- | | | |
|--|-------|---------------|
| 1. Hendra Cordova, S.T., M.T | | Pembimbing |
| 2. Dr. Ir. Totok Soehartanto, DEA | | Ketua Penguji |
| 3. Dr. Katherin Indriawati, S.T., M.T. | | Penguji I |

SURABAYA
Juli 2017

RANCANG BANGUN SISTEM PENGENDALI KADAR OKSIGEN TERLARUT DENGAN ALGORITMA *FUZZY LOGIC CONTROLLER* PADA BUDIDAYA AKUAPONIK

Nama Mahasiswa : Alberto Riolly Cahyantara
NRP : 2412 100 060
Departemen : Teknik Fisika
Dosen Pembimbing : Hendra Cordova, S.T., M.T.

Abstrak

Dalam budidaya akuaponik, kadar oksigen terlarut sangat penting karena mempengaruhi laju pertumbuhan ikan dan tanaman. Untuk menjaga kadar oksigen pada standar baku mutu air (set point), maka laju aerasi harus dimanipulasi dengan mengatur kerja pompa aerator. Langkah awal adalah dengan mengetahui karakteristik pompa aerator terhadap peningkatan kadar oksigen terlarut pada akuaponik. Selanjutnya algoritma *fuzzy logic controller* dibuat dan digunakan untuk menentukan variabel manipulasi dari pompa aerator pada akuaponik. Dengan menggunakan algoritma *Fuzzy logic controller*, kontrol kadar oksigen terlarut dapat dikontrol dan didapatkan nilai *maximum overshoot* (Mp) 1.27%, *rise time* (t_r) 211 detik, *settling time* (t_s) 575 detik, *ess* 0.07 mg/L atau 1%. *Fuzzy logic controller* mampu mengatasi gangguan dan mengikuti perubahan *set point* yang diberikan. Dengan kontrol kadar oksigen terlarut pada akuaponik, terjadi kenaikan laju pertumbuhan ikan sebesar 0.21 gram dan peningkatan tinggi tanaman sebesar 1.2 cm selama 10 hari pengamatan.

Kata kunci: oksigen terlarut (dissolved oxygen), *fuzzy logic controller* (FLC), akuaponik, laju pertumbuhan

Halaman ini memang dikosongkan

DESIGN OF DISSOLVED OXYGEN CONTROL SYSTEM USING FUZZY LOGIC CONTROLLER ALGORITHM ON AQUAPONIC

Name : Alberto Riolly Cahyantara
Student Number : 2412 100 060
Department : Engineering Physics
Supervisor : Hendra Cordova, S.T., M.T.

Abstract

In aquaponics, dissolved oxygen levels are important because they affect the growth rate of fish and plants that are cultivated. To maintain the dissolved oxygen level at the water quality standard (set point), the aeration rate must be manipulated by regulating work of aerator pump. The first step is to understand the aerator pump characteristic on the effect of dissolved oxygen levels on the aquaponics. Then, the fuzzy logic controller algorithm is created and used to determine the manipulation variables of aerator pumps on aquaponics. By using Fuzzy Logic Controller algorithm, the control performance of dissolved oxygen level obtained the maximum overshoot (Mp) 1.27%, rise time (t_r) 211 second, settling time (t_s) 574 second, ess 0.07 mg/L or 1%. The controller is able to overcome disturbance and tracking the change of set point that are given. With the controlling dissolved oxygen levels in aquaponics, result in increasing the fish growth rate by 0.21 gram or and plants height by 1.2 cm during ten days of experiment.

Key Words: *dissolved oxygen (DO), fuzzy logic controller (FLC), aquaponic, fish and vegetables growth rate*

Halaman ini memang dikosongkan

KATA PENGANTAR

Puji syukur dan segala kemuliaan bagi Yesus Kristus, Tuhan Yang Maha Esa, yang telah memberikan hikmat dan kekuatan kepada penulis sehingga penulis mampu menyelesaikan penelitian tugas akhir dengan judul:

RANCANG BANGUN SISTEM PENGENDALI KADAR OKSIGEN TERLARUT DENGAN ALGORITMA FUZZY LOGIC CONTROLLER PADA BUDIDAYA AKUAPONIK

Penyusunan laporan penelitian tugas akhir ini tidak terlepas dari semua pihak yang turut membantu baik moril maupun materiil. Untuk itu penulis mengucapkan terima kasih kepada:

1. Orang tua saya, Bapak Siswadi dan Ibu Naryati, serta saudara-saudara saya, Yulius, Nathan, dan Michael yang selalu mendukung dalam doa dan motivasi untuk kebaikan kelancaran tugas akhir ini.
2. Bapak Agus Muhammad Hatta, S.T., M.Si, Ph.D selaku Kepala Departemen Teknik Fisika ITS
3. Bapak Hendra Cordova, S.T., M.T., dan Bapak Totok Ruki Biyanto, Ph.D, selaku dosen yang dengan sabar membimbing, memberikan saran dan kritiknya yang membangun.
4. Bapak Herry dari Roxtrade Industries Co. Ltd.yang telah mendukung dalam proses pembuatan alat untuk penelitian tugas akhir ini.
5. Teman-teman Teknik Fisika Angkatan berapapun, terutama Zendy, Dimas, Ichal, Fernando, serta teman-teman Asisten Laboratorium Rekayasa Instrumentasi dan Kontrol dan kader-kader HMTF ITS, sahabat yang senantiasa memberikan semangat.

Demikian laporan tugas akhir ini dibuat dengan sebaik-baiknya. Dalam penelitian tugas akhir dan penyusunan laporan ini, masih banyak kesalahan yang harus diperbaiki. Maka dari itu, penulis mengharapkan kritik dan saran untuk perbaikan

penelitian selanjutnya. Semoga laporan ini bermanfaat bagi semua pihak, khususnya untuk kemajuan pertanian dan perikanan di Indonesia.

Surabaya, Juli 2017
Penulis

DAFTAR ISI

HALAMAN SAMPUL	i
LEMBAR PENGESAHAN	vii
ABSTRAK.....	xi
KATA PENGANTAR	xiii
DAFTAR GAMBAR.....	xix
DAFTAR TABEL.....	xxi
BAB I PENDAHULUAN.....	1
1.1. Pendahuluan.....	1
1.2. Rumusan Masalah	3
1.3. Tujuan	3
1.4. Lingkup Pekerjaan	4
1.5. Manfaat	4
BAB II DASAR TEORI	5
2.1. Budidaya Akuaponik.....	5
2.1.1. Kualitas Air.....	6
2.1.2. Ikan Nila	7
2.1.3. Selada.....	8
2.1.4. Pengukuran Laju Pertumbuhan.....	9
2.2. Dissolved Oxygen (DO).....	10
2.2.1. Pengukuran DO.....	13
2.2.2. Persen Saturasi Oksigen.....	14
2.2.3. Sistem Pengendali DO	15
2.3. Komponen Sistem Pengendali	16
2.3.1. Mikrokontroler	16
2.3.2. Sensor DO	17

2.3.3. Rangkaian Pengkodisian Sinyal.....	18
2.3.4. Pompa Aerator	19
2.3.5. Relay	20
2.3.6. Catu Daya.....	20
2.4. Logika Fuzzy.....	21
BAB III METODOLOGI PENELITIAN	25
3.1. Alur Penelitian	25
3.2. Studi Literatur	26
3.3. Perancangan Akuaponik	27
3.4. Perancangan Sistem Kontrol DO	28
3.5. Pengujian Karakteristik Sistem.....	29
3.6. Perancangan Algoritma Logika Fuzzy.....	31
3.7. Pengujian Performa Sistem Kontrol.....	35
3.8. Perancangan Interface Monitoring dan Kontrol.....	36
3.9. Implementasi Sistem Kontrol.....	37
BAB IV ANALISA DATA DAN PEMBAHASAN	39
4.1. Uji Karakteristik Alat.....	39
4.1.1 Kalibrasi Sensor	39
4.2. Uji Open Loop Sistem.....	40
4.3. Uji Closed Loop Sistem	41
4.4. Laju Pertumbuhan Akuaponik	46
BAB V PENUTUP	51
5.1. Kesimpulan	51
5.2. Saran	51
LAMPIRAN A Hasil Perancangan Mekanik dan Elektrik	
LAMPIRAN B Program Fuzzy Pada Arduino	
LAMPIRAN C <i>Source Code Firmware</i>	

Daftar Gambar

Gambar 2.1 Siklus Budidaya Akuaponik (abundance.org)	6
Gambar 2.2 Ikan Nila (tandapagar.com, 2016)	8
Gambar 2.3 Sayur Selada (bibitbunga, 2017)	9
Gambar 2.4 Grafik Pengaruh Konsentrasi DO terhadap Laju Pertumbuhan, Kelulushidupan, dan FCR pada Ikan (Mallya, 2007).....	11
Gambar 2.5 Fluktuasi Harian Kadar DO (Edis, 2013)	12
Gambar 2.6 Hubungan DO dan Temperature (CoTF, 2004).....	12
Gambar 2.7 Sensor DO Galvanik (Fondriest, 2014)	13
Gambar 2.8 Diagram Blok Sistem Pengendalian Kadar DO (Riolly, 2017).....	16
Gambar 2.9 Arduino Mega (Arduino, 2013).....	16
Gambar 2.10 Sensor DO (AtlasScientific, 2015)	18
Gambar 2.11 Rangkaian Pengkondisi Sinyal Sensor DO (AtlasScientific, 2015).....	18
Gambar 2.12 Pompa Aerator TJ-2600 (Aquacave, 2017)	19
Gamber 2.13 Relay (Omron, 2015)	20
Gambar 2.14 Catu Daya (Tachopedia, 2017)	20
Gambar 2.15 Arsitektur logika <i>fuzzy</i> Mamdani (Wang, 1997).....	21
Gambar 2.16 Fungsi Keanggotaan Segitiga	22
Gambar 2.17 Fungsi Keanggotaan Trapesium	23
Gambar 2.18 Representasi Kurva <i>Centroid</i>	24
Gambar 3.1 Diagram Alir Tahapan Penelitian	25
Gambar 3.2 Desain Kolam Akuaoponik	27
Gambar 3.3 Skematik Rangkaian Alat Pengendali Kadar DO	28
Gambar 3.4 Karakteristik Respon DO Akibat Pompa Aerator	30
Gambar 3.5 Respon DO Ketika Dinyalakan Pompa Aerator Sebanyak (a) 1 Pompa, (b) 2 Pompa, dan (c) 3 Pompa Sampai Mencapai Keadaan Jenuh.	31
Gambar 3.6 FIS Sistem Pengendali DO	32
Gambar 3.7 Fungsi Keanggotaan <i>error</i> DO	33

Gambar 3.8 Fungsi Keanggotaan % Saturasi	33
Gambar 3.9 Fungsi Keanggotaan Pompa Aerator	34
Gambar 3.10 Aturan Logika Fuzzy	35
Gambar 3.11 Pemetaan <i>Input Output</i> FIS	35
Gambar 3.12 Air Limbah Budidaya sebagai Gangguan.....	36
Gambar 3.13 <i>Interface</i> Sistem <i>Monitoring</i> dan Kontrol DO.....	37
Gambar 4.1 Kalibrasi Sensor DO dengan YSI 500A	39
Gambar 4.2 Perubahan DO disebabkan oleh ikan	40
Gambar 4.3 Perubahan DO disebabkan oleh pompa	41
Gambar 4.4 Tracking set point DO dari 4.5 mg/L menuju 5,5 mg/L, menuju 6 mg/L.....	42
Gambar 4.5 Respon DO terhadap disturbance 10 L.....	43
Gambar 4.6 Respon DO terhadap disturbance 20 L.....	44
Gambar 4.7 Respon DO terhadap disturbance 30 L.....	45
Gambar 4.8 Perbandingan respon DO dengan fuzzy dan on/off (4 pompa)	46
Gambar 4.9 Perbandingan Ikan dengan DO Dikendalikan (kanan) dan Tidak (kiri).....	47
Gambar 4.10 Perbandingan sayur dengan DO dikendalikan (kiri) dan tidak (kanan)	49

Daftar Tabel

Tabel 2.1 Standar Kualiatas Air Ikan Nila (Badan Standardisasi Nasional, 2009)	8
Tabel 2.2 Spesifikasi Arduino Mega	17
Tabel 2.3 Spesifikasi Sensor DO	17
Tabel 2.4 Spesifikasi Rangkaian Pengkondisi Sinyal (AtlasScientific, 2015)	18
Tabel 3.1 Hubungan <i>Input-Output</i> berdasarkan <i>Rule Base</i>	34
Tabel 4.1 Hasil Kalibrasi Sensor DO	40
Tabel 4.2 Performa Kontrol saat Tracking Set Point	42
Tabel 4.3 Perbandingan Laju Pertumbuhan Ikan	47
Tabel 4.4 Perbandingan Laju Pertumbuhan Tanaman	48

BAB I

PENDAHULUAN

1.1. Pendahuluan

Permasalahan yang dihadapi dunia saat ini adalah meningkatnya populasi manusia yang berakibat pada tingginya kebutuhan pangan. Diperkirakan pada tahun 2050, kebutuhan pangan meningkat 69% untuk 6.9 milyar penduduk dunia. (World Resources Institute, 2015) Untuk mencukupi kebutuhan pangan tersebut, telah dilakukan intensifikasi produksi pangan pada sektor agrikultur. Intensifikasi produksi berarti mengeksploitasi tanah dan air secara berlebihan yang telah berakibat pada degradasi tanah dan kelangkaan air (Food and Agriculture Organization of the United Nations, 2011). Untuk menghadapi permasalahan ini, dibutuhkan proses produksi pangan yang berkelanjutan dan meminimalisir efek negatif pada lingkungan.

Akuakultur atau budidaya perikanan merupakan sektor produksi pangan paling cepat berkembang dan telah berhasil memenuhi setengah dari konsumsi ikan di dunia (Food and Agriculture Organization of United Nations, 2016). Percepatan peningkatan produksi akuakultur merupakan hasil dari intensifikasi. Intensifikasi akuakultur disisi lain, telah menimbulkan efek negatif karena menggunakan banyak sumber air permukaan yang jumlahnya terbatas. Limbah yang dikeluarkan juga akan merusak kualitas air dan mengancam organisme ditempat pembuangan (Jegatheesan, Shu, & Visvanathan, 2011). Hal ini kemudian diatasi dengan menggunakan *recirculating aquaculture system* (RAS) yang mampu mengurangi 90% penggunaan air dan mengurangi limbah yang dibuang ke lingkungan (Timmons, Ebeling, Wheaton, Summerfelt, & Vinci, 2007).

Akuaponik merupakan gabungan dari produksi ikan secara intensif dalam *recirculating aquaculture system* (RAS) dengan produksi tanaman hidroponik (Rakocy & Hargreaves, 1993). Pada RAS terjadi akumulasi unsur organik yang tidak diinginkan ikan (limbah) yang kebanyakan adalah unsur nitrogen berupa amonia. Amonia ini diurai oleh bakteri nitrifikasi dan dimanfaatkan sebagai

nutrisi untuk pertumbuhan tanaman. Selain menghilangkan limbah, produksi tanaman akan menjadi keuntungan tambahan pada sistem akuaponik.

Akuaponik atau sering disebut *single recirculating aquaponic system* (SRAPS) merupakan sistem yang sangat kompleks. Hal ini dikarenakan tiga organisme yaitu ikan, tanaman, dan bakteri nitrifikasi digabungkan dalam satu sistem. Sedangkan masing-masing organisme membutuhkan kondisi kualitas air yang berbeda-beda supaya pertumbuhannya/ fungsinya optimal seperti kestabilan pH air, kesetimbangan nutrisi, kandungan fosfor, dan pengendalian hama (Goddek, Delaide, Mankasingh, & Ragnarsdottir, 2015). Kadar oksigen terlarut (DO) juga menjadi salah satu variabel yang menjadi faktor yang mempengaruhi hasil produksi ikan dan sayuran serta kinerja dari bakteri nitrifikasi dalam melakukan penguraian.

Pada budidaya ikan, contohnya ikan nila, kenaikan berat akhir secara signifikan terjadi dengan menaikkan konsentrasi oksigen terlarut. (Papoutsoglou & Tziha, 1996). Pada kondisi *hypoxia* atau kadar oksigen kurang (3.5 mg/l) kemampuan ikan mencerna nutrisi menurun drastis (Tran-Ngoc, et al., 2016). Dalam budidaya udang vaname, kenaikan kadar oksigen terlarut dalam kolam meningkatkan tingkat kelulushidupan dan keuntungan yang diperoleh (McGraw, Teichert-Coddington, Rouse, & Boyd, 2001). Kenaikan kadar oksigen terlarut juga menurunkan *food conversion ratio* (FCR) (Papoutsoglou & Tziha, 1996). Sehingga kadar oksigen dalam budidaya akuaponik perlu dikontrol pada kondisi yang optimal untuk menghasilkan hasil panen yang optimal.

Pada sisi tanaman hidroponik, kekurangan oksigen pada akar tanaman akan menyebabkan pertumbuhan akar dan tanaman menjadi buruk serta meningkatnya resiko terkena penyakit (Cherif, Tirilly, & Belanger, 1997). Laju pertumbuhan, berat akar, bentuk akar, luas daun, laju konsumsi air, dan hasil panen menurun dengan kadar oksigen yang rendah pada larutan nutrisi (Gislerod & Kempton, 1983). Pada *aero-hydroponics* juga terjadi penurunan serupa (Soffer, Burger, & Lieth, 1991).

Kadar oksigen terlarut biasa dikendalikan menggunakan sistem kontrol modern pada instalasi pengolahan air limbah industri untuk

mengoptimalkan proses nitrifikasi oleh bakteri. Metode kontrol yang sering digunakan dalam sistem kontrol kadar oksigen terlarut pada pengolahan air limbah adalah *fuzzy*. *Fuzzy* menunjukkan performa yang lebih baik untuk pengendalian DO pada pengolahan air limbah, pada proses *activated sludge* (Amand, Olsson, & Carlsson, 2013), dan juga tipe *sequencing batch reactor* (SBR) (Piotrowski & Skiba, 2015). Karena karakteristik dinamik kadar oksigen terlarut tidak linier dan dipengaruhi oleh banyak *disturbance* seperti temperatur (Seginer & Mozes, 2012), metode *fuzzy* lebih tepat dipakai untuk mengendalikan kadar oksigen terlarut dibandingkan metode kontrol lain.

Maka dari itu, dalam upaya pengembangan penelitian dibidang akuaponik, dibuat sistem kontrol kadar oksigen terlarut pada budidaya akuaponik. Pada sistem, diimplementasikan logika *fuzzy* sebagai metode kontrol sistem. Penelitian dilakukan secara eksperimen, yaitu dengan membuat rancang bangun *single recirculating aquaponic system* (SRAPS) beserta komponen kontrolnya. Harapannya sistem kontrol yang dibuat tidak hanya meningkatkan hasil produksi ikan dan tanaman, namun juga mempermudah kerja petani dalam manajemen akuaponik.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang sudah dijelaskan, dapat dirumuskan masalah dalam penelitian tugas akhir ini adalah:

- 1) Bagaimana merancang sistem kontrol kadar oksigen terlarut pada sistem akuaponik dengan mengimplementasikan metode kontrol logika *fuzzy*?
- 2) Bagaimana pengaruh penggunaan sistem kontrol kadar oksigen terlarut terhadap pertumbuhan ikan dan tanaman akuaponik?

1.3. Tujuan

Dari rumusan masalah yang didapat, maka tujuan dari penelitian tugas akhir ini adalah:

- 1) Merancang sistem kontrol kadar oksigen terlarut pada sistem akuaponik dengan mengimplementasikan metode kontrol logika *fuzzy*.

- 2) Mengetahui pengaruh penggunaan sistem kontrol kadar oksigen terlarut terhadap pertumbuhan ikan dan tanaman akuaponik.

1.4. Lingkup Pekerjaan

Adapun lingkup penelitian tugas akhir yang dikerjakan meliputi:

- 1) Jenis akuaponik yang dibuat adalah *single recirculating aquaponic system* (SRAPS) jenis *media filled bed* dengan pasir malang sebagai media tanamnya.
- 2) Jenis ikan dan sayuran yang dibudidaya adalah ikan nila dan sawi.
- 3) Pengaruh pengendalian kadar oksigen terhadap laju pertumbuhan ikan dan tanaman dilakukan selama 10 hari.
- 4) *Electro conductivity* (EC) atau kadar nutrisi air sebagai variabel kontrol penelitian yaitu dibuat tetap dan tidak diamati pengaruhnya.

1.5. Manfaat

Manfaat yang didapatkan dari penelitian tugas akhir ini adalah:

- 1) Membantu dan mempermudah kerja petani akuaponik dalam melakukan *monitoring* dan menjaga kualitas air dari akuaponik miliknya.
- 2) Meningkatkan hasil panen dari budidaya akuaponik sehingga meningkatkan keuntungan petani.

BAB II

TINJAUAN PUSTAKA

2.1. Budidaya Akuaponik

Akuaponik merupakan kombinasi dari teknik budidaya akuakultur dengan teknik budidaya tanaman secara hidroponik. Akuaponik merupakan suatu sistem tanam terpadu yang menggunakan hasil perombakan bahan organik (limbah) dalam budidaya ikan sebagai nutrisi bagi budidaya tanaman.

Pada akuakultur, pertumbuhan ikan dimaksimalkan dalam kolam atau wadah pemeliharaan yang terbatas. Pada sistem seperti ini, ikan ditebar dalam jumlah yang cukup banyak dalam kolam atau wadah pemeliharaan yang terbatas sehingga memiliki tingkat kepadatannya tinggi. (Sari, 2014) Hal ini mengakibatkan tingginya tingkat limbah organik dalam air pemeliharaan ikan, sehingga menyebabkan kualitas air akuakultur akan turun dan pada akhirnya dapat meracuni ikan.

Budidaya tanaman dengan cara hidroponik membutuhkan nutrisi dan unsur-unsur mikro yang harus ditambahkan dalam media tanam. Kebutuhan nutrisi, unsur mikro, dan unsur organik tersebut dibuat dan ditambahkan agar tanaman dapat tumbuh dengan optimal. Tanpa menambahkan nutrisi berupa pupuk cair, tanaman hidroponik tidak bisa tumbuh dengan normal dan hasilnya tidak optimal.

Dalam akuaponik, limbah organik yang dihasilkan oleh ikan akan diurai oleh bakteri nitrifikasi yang akan mengubah amonia menjadi nitrat. Nitrat inilah yang akan menjadi nutrisi bagi tanaman hidroponik. Prinsip akuaponik adalah resirkulasi air, dimana air dari pemeliharaan ikan secara terus menerus dialirkan ke media tanam tanaman untuk disaring dan kembali lagi ke air pada pemeliharaan ikan (Sari, 2014).

Akuaponik dapat dilakukan di lahan yang sempit. Dengan teknik resirkulasi air dapat mengurangi pemakaian air untuk budidaya ikan, mengurangi pencemaran limbah budidaya ikan, dan memastikan ketersediaan air bersih untuk budidaya ikan (Sari, 2014). Namun sama seperti budidaya ikan dan budidaya hidroponik,

kualitas air pada sistem akuaponik juga memiliki baku mutu atau nilai standar supaya hasilnya maksimal. Di dalam akuaponik nilai kualitas air harus presisi karena harus menyesuaikan baku mutu yang dibutuhkan tanaman dan ikan yang memiliki rentang nilainya tidak sama.



Gambar 2.1 Siklus Budidaya Akuaponik (abundance.org)

2.1.1. Kualitas Air

Berdasarkan peraturan pemerintah no 82 tahun 2001, Baku mutu air adalah ukuran batas atau kadar makhluk hidup, zat, energi, atau komponen yang ada atau harus ada dan atau unsur pencemar yang ditenggang keberadaannya di dalam air. Suatu perairan apabila telah memenuhi baku mutu yang ditentukan sesuai peruntukannya maka perairan tersebut dapat digolongkan kualitasnya baik digunakan, sesuai peruntukannya. Kondisi kualitas air dapat diukur dan atau diuji berdasarkan parameter-parameter dan metoda tertentu. Parameter kualitas air pada proses budidaya ikan berperan dalam menciptakan suasana lingkungan hidup ikan, agar perairan kolam mampu memberikan suasana yang nyaman bagi pergerakan ikan yaitu tersedianya air yang cukup untuk menciptakan kualitas air yang sesuai dengan persyaratan hidup ikan yang optimal (kimia, fisika, dan biologi) sesuai dengan parameter yang disyaratkan.

Parameter kualitas air untuk budidaya ikan air tawar meliputi 3 karakteristik, yaitu karakteristik fisik, kimia serta karakteristik biologi (plankton). Faktor-faktor penting kualitas air yang perlu mendapat perhatian diantaranya adalah suhu air, salinitas, oksigen terlarut, pH, alkalinitas, ammonia, nitrit, nitrat, asam sulfida, karbondioksida, dan besi. Faktor-faktor tersebut dalam suatu tempat terus mengalami perubahan dinamis karena adanya faktor di luar dan di dalam sistem yang kemudian saling mempengaruhi antar faktor tersebut. Perubahan lingkungan secara kimia dan fisika bisa terjadi secara alamiah atau akibat perbuatan manusia.

2.1.2. Ikan Nila

Ikan nila (*Oreochromis niloticus*) merupakan jenis ikan air tawar yang mempunyai nilai konsumsi cukup tinggi. Bentuk tubuh memanjang dan pipih ke samping dan warna putih kehitaman atau kemerahan. Ikan nila berasal dari Sungai Nil dan danau-danau sekitarnya. Ikan nila umum hidup di perairan tawar, terkadang ikan nila juga ditemukan hidup di perairan yang agak asin (payau). Ikan nila dikenal sebagai ikan yang bersifat *euryhaline* (dapat hidup pada kisaran salinitas yang lebar). Ikan nila mendiami berbagai habitat air tawar, termasuk saluran air yang dangkal, kolam, sungai dan danau. Ikan nila tidak mampu untuk bertahan hidup di perairan dingin, yang umumnya bersuhu di bawah 21 °C. Ikan Nila adalah termasuk ikan omnivora (Mudjiman, 2001),

Ikan nila mempunyai kemampuan tumbuh secara normal pada kisaran suhu 14-38°C dengan suhu optimum bagi pertumbuhan dan perkembangannya yaitu 25-30°C. Pada suhu 14°C atau pada suhu tinggi 38°C pertumbuhan ikan nila akan terganggu. Pada suhu 6°C atau 42°C ikan nila akan mengalami kematian. Kandungan oksigen yang baik bagi pertumbuhan ikan nila minimal 4 mg/L, kandungan karbondioksida kurang dari 5mg/L dengan derajat keasaman (pH) berkisar 5-9 (Amri & Khairuman, 2003). Peningkatan berat akhir ikan nila secara signifikan terjadi dengan menaikkan konsentrasi oksigen terlarut. (Papoutsoglou & Tziha, 1996). Pada kondisi *hypoxia* atau kadar oksigen kurang (3.5 mg/l) kemampuan ikan mencerna nutrisi menurun drastis (Tran-Ngoc, et al., 2016).

Menurut (Santosa, 2001), pH optimum bagi pertumbuhan nila yaitu antara 7-8 dan warna di sekujur tubuh ikan dipengaruhi lingkungan hidupnya. Kandungan CO_2 dalam air untuk kegiatan pembesaran nila sebaiknya kurang dari 15 mg/liter (Sucipto & Prihartono, 2005). Jumlah CO_2 di dalam perairan yang bertambah akan menekan aktivitas pernapasan ikan dan menghambat pengikatan oksigen oleh hemoglobin sehingga dapat membuat ikan menjadi stres. Nilai CO_2 ditentukan antara lain oleh pH dan suhu. Di sisi lain, konsentrasi karbondioksida diperlukan untuk proses fotosintesis oleh tanaman air.



Gambar 2.2 Ikan Nila (tandapagar.com, 2016)

Menurut Badan Standar Nasional Indonesia, standar kualitas air untuk budidaya ikan nila adalah sebagai berikut.

Tabel 2.1 Standar Kualitas Air Ikan Nila (Badan Standardisasi Nasional, 2009)

No	Parameter	Satuan	Nilai
1	Suhu	C	25-32
2	pH	-	6.5-8.5
3	Oksigen terlarut	mg/l	> 5
4	Amoniak	mg/l	< 0.02
5	Kecerahan	cm	30-40

2.1.3. Selada

Selada (*Lactuca sativa* L) adalah tanaman yang termasuk dalam famili Compositae (Sunarjono, 2014). Selada merupakan tanaman setahun yang dapat di budidayakan di daerah lembab, dingin, dataran rendah maupun dataran tinggi. Pada dataran tinggi yang beriklim lembab produktivitas selada cukup baik. Selada merupakan sayuran

dengan nilai ekonomis tinggi yang populer dijadikan salad atau dimakan mentah.

Suhu ideal untuk produksi selada berkualitas tinggi adalah 15-25 °C. Suhu yang lebih tinggi dari 30°C dapat menghambat pertumbuhan, merangsang tumbuhnya tangkai bunga (bolting), dan dapat menyebabkan rasa pahit. Syarat penting agar selada dapat tumbuh dengan baik yaitu memiliki derajat keasaman tanah pH 5-6.5 (Sunarjono, 2014). Selada dapat tumbuh pada jenis tanah lempung berdebu, berpasir dan tanah yang masih mengandung humus ataupun secara hidroponik. Pada budidaya secara hidroponik, selain pH yang harus dikontrol, kandungan oksigen terlarut juga mempengaruhi pertumbuhan selada. Pada konsentrasi DO yang rendah, akar tanaman tidak bisa menyerap nutrisi dengan baik, sehingga menurunkan pertumbuhan, menyebabkan berat daun selada menurun (Yoshida, Kitano, & Eguchi, 1997).



Gambar 2.3 Sayur Selada (bibitbunga, 2017)

2.1.4. Pengukuran Laju Pertumbuhan

Untuk mengetahui laju pertumbuhan dari suatu organisme yang dibudidayakan, ada beberapa parameter yang bisa digunakan sebagai acuan. Namun secara umum, pertumbuhan diukur dengan membandingkan berat awal dengan berat akhir, baik untuk laju pertumbuhan ikan maupun laju pertumbuhan tanaman. Laju pertumbuhan dinyatakan sebagai perubahan bobot tubuh rata-rata selama percobaan berlangsung (Haliman, 2015). Rumus untuk menghitung laju pertumbuhan (GR) adalah sebagai berikut.

$$Gr = \frac{W_t - W_o}{t} \quad (2.1)$$

GR = Laju pertumbuhan

Wt = Berat rata-rata organisme akhir percobaan (g)

Wo = Berat rata-rata organisme awal percobaan (g)

t = Jumlah hari selama percobaan

Selain perubahan berat, laju pertumbuhan juga bisa diukur dengan membandingkan panjang akhir dan awal. Untuk ikan, tingkat kelulushidupan juga menjadi salah satu parameter keberhasilan yang penting yaitu perbandingan berapa banyak ikan yang ditabur dengan yang di panen. Untuk tanaman, lebar dan jumlah daun serta besarnya akar menjadi salah satu indikator suburnya tanaman.

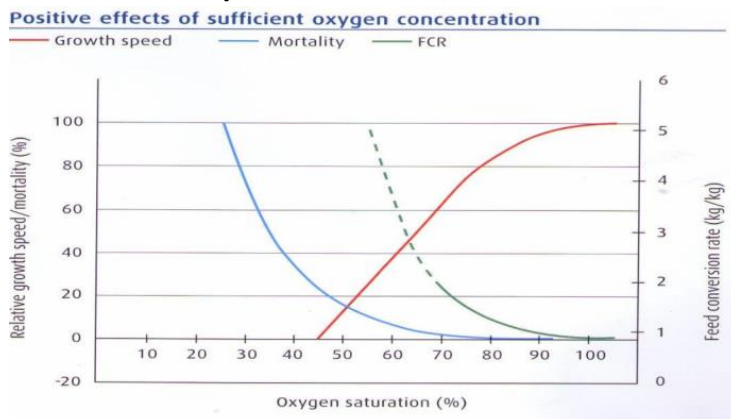
2.2. Dissolved Oxygen (DO)

Sebagai makhluk hidup, ikan dan hewan air lainnya memerlukan oksigen terlarut di dalam air untuk respirasi dan proses metabolisme. Kebutuhan oksigen untuk setiap jenis ikan sangat berbeda karena perbedaan sel darahnya. Ikan yang gesit seperti ikan nila umumnya lebih banyak membutuhkan oksigen. Sementara ikan labirin seperti lele, *catfish*, dan gurame yang dapat mengambil oksigen langsung dari udara, mampu hidup diperairan dengan kadar oksigen rendah. Secara teori, kadar oksigen terlarut terendah agar ikan bisa hidup dengan baik adalah lebih dari 5 mg/l. Namun demikian pada konsentrasi yang berlebihan dapat mengakibatkan kematian dengan terjadinya emboli dalam pembuluh darah akibat terlalu banyak gelembung udara (gas bubble disease).

Kebutuhan oksigen ikan dan organisme lainya juga sangat bergantung pada faktor-faktor suhu, pH, CO₂, dan kecepatan metabolik ikannya. Faktor pentingnya adalah suhu air dan berat tubuh. Kebutuhan oksigen meningkat dengan meningkatnya suhu air. Berat tubuh akan berkaitan dengan aktivitas dan akan meningkatkan respirasi. Semakin tinggi berat akan semakin menurun kebutuhan oksigennya.

Metabolik ikan akan berkurang/berhenti ketika suhu tidak optimum atau perubahannya terlalu ekstrim. Jika suhu air meningkat maka jumlah kandungan oksigen menurun dan konsumsi oksigen oleh ikan dan organisme di dalam air meningkat. Oksigen yang berkurang berdampak pada aktivitas ikan berkurang atau berhenti karena nafsu makannya berhenti. Makanan akan tersisa dan

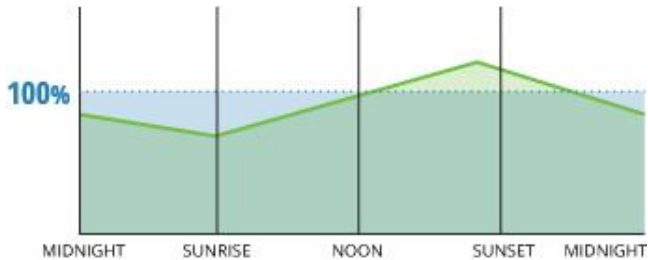
berdampak pada meningkatnya akumulasi amoniak di air yang berakibat pada munculnya serangan penyakit. Kadar oksigen yang cukup akan meningkatkan pertumbuhan ikan, mengurangi tingkat kematian pada ikan, dan menurunkan *food conversion ratio* (FCR), yaitu seberapa besar makanan yang dikonversi menjadi daging ikan (semakin kecil nilainya semakin efisien).



Gambar 2.4 Grafik Pengaruh Konsentrasi DO terhadap Laju Pertumbuhan, Kelulushidupan, dan FCR pada Ikan (Mallya, 2007)

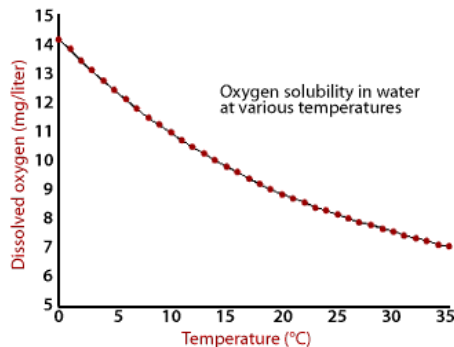
Pada lingkungan perairan, ada tiga sumber oksigen yaitu: 1) difusi dari atmosfer, 2) pergerakan angin dan ombak, dan 3) fotosintesis. Fotosintesis menjadi pemasok oksigen utama yang dilakukan oleh tumbuhan air dan fitoplankton (Francis & Floyd, 2014).

Oksigen, dihasilkan dari proses fotosintesis, diproduksi selama siang hari dimana ada sinar matahari yang mengenai tanaman air dan fitoplankton. Kadar oksigen terlarut akan menurun pada saat malam karena digunakan untuk respirasi organisme air termasuk ikan tanpa adanya suplai oksigen dari fotosintesis. Dekomposisi yang dilakukan bakteri pengurai juga mengkonsumsi oksigen dalam air. Fluktuasi oksigen ini bisa diprediksi, terjadi selama 24 jam, disebut sebagai siklus oksigen harian.



Gambar 2.5 Fluktuasi Harian Kadar DO (Edis, 2013)

Kekurangan oksigen akan menyebabkan ikan sesak napas, aktivitas kurang dan ikan akan mati. Ikan yang kekurangan oksigen tidak akan mengambil makanan, proses metabolismenya terbatas. Pada kondisi tersebut, mereka akan mencari tempat yang mempunyai oksigen banyak seperti ke permukaan air atau sekitar pemasukan air. Pengendalian masalah kekurangan oksigen di dalam air adalah dengan memberikan aerasi melalui mesin pemompa udara atau menggunakan kincir air agar air bergerak untuk memudahkan proses penyerapan oksigen bebas oleh air.



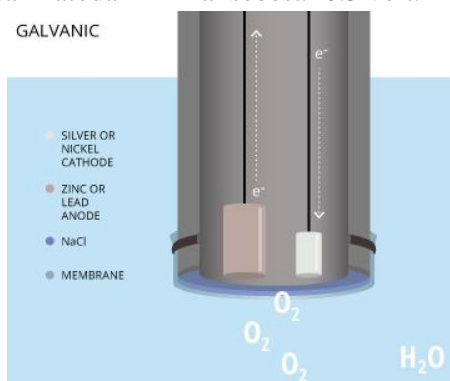
Gambar 2.6 Hubungan DO dan Temperature (CoTF, 2004)

Kemampuan oksigen untuk terlarut pada air sangat dipengaruhi oleh temperatur air. Semakin tinggi temperatur semakin menurun kecil kadar oksigen yang mampu terlarut dalam air seperti ditunjukkan oleh Gambar 2.6.

2.2.1. Pengukuran DO

Salah satu metode pengukuran kadar oksigen terlarut adalah dengan prinsip *electrochemical* dengan sensor galvanik. Sebuah logam memiliki beda potensial listrik berdasarkan urutan reaktivitas unsurnya (seberapa cepat memberi dan menerima elektron) (Ladon, 2001). Ketika ditempatkan pada larutan elektrolit, potensial antara dua buah logam yang berbeda menyebabkan kedua logam terpolarisasi (Finesse, LLC, n.d.).

Anoda pada sensor ini adalah seng, atau menggunakan logam aktif, sedangkan katoda menggunakan silver, atau logam mulia (Hargreaves & Tucker, 2002). Larutan elektrolit berupa sodium hidroksida, sodium klorida, atau elektrolit inert lainnya (Eutech Instruments, 1997) (Wilkin & Ptacek, 2000). Kedua logam elektroda akan terpolarisasi, dengan elektron mengalir dari anoda ke katoda. Katoda tetap *inert*, hanya memberikan jalan bagi elektron dan tidak mempengaruhi reaksi (YSI, 2009). Anoda teroksidasi dan oksigen tereduksi pada permukaan katoda. Agar mampu mereduksi oksigen tanpa memerlukan tegangan eksternal, dibutuhkan beda potensial antara anoda dan katoda minimal sebesar 0.5 volt.



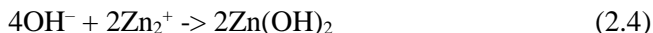
Gambar 2.7 Sensor DO Galvanik (Fondriest, 2014)

Reaksi yang terjadi adalah sebagai berikut.

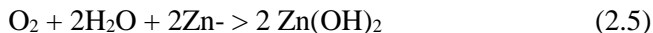
1. Reaksi Anoda Seng and Oksidasi.



2. Reaksi Katoda Silver dan Reduksi Oksigen.



Sehingga, keseluruhan reaksi yang terjadi sebagai berikut.



Dalam reaksi polarografi, katoda tidak masuk dalam persamaan karena bersifat *inert* (Larsen, 2014). Katoda silver menerima elektron dari anoda dan mentransfernya ke molekul oksigen. Perpindahan terjadi pada permukaan katoda (Watt, 2000). Arus yang dihasilkan dengan reduksi oksigen proporsional dengan tekanan parsial dari oksigen pada sampel air yang diukur (Thermo Electron Corporation, n.d.).

Seng hidroksida dihasilkan dari reaksi kimia, mengendap dalam larutan elektrolit. Endapan akan terlihat berbentuk solid berwarna putih di ujung sensor (YSI, 2009). Endapan ini tidak akan menutupi anoda atau menghabiskan elektrolit, sehingga tidak mempengaruhi performa sensor. Namun ketika jumlah endapan sudah cukup banyak, kemampuan ion untuk menghantarkan listrik antara anoda dan katoda akan berkurang (YSI Incorporated, 2008). Saat keluaran sensor rendah dan pembacaan tidak stabil, maka elektrolit perlu diganti.

Ketika elektroda sensor galvanik terpolarisasi, oksidasi seng akan terus terjadi walaupun sensor tidak digunakan. Galvanik sensor akan tetap berfungsi efisien walaupun seng terus bereaksi, hal ini membuat sensor galvanik akan lebih sering diganti dibanding sensor DO jenis polarografi.

2.2.2. Persen Saturasi Oksigen

Persen saturasi oksigen adalah perbandingan konsentrasi oksigen yang diamati dengan kelarutan oksigen absolut di air pada temperatur tertentu. Persamaan persen saturasi oksigen sebagai berikut (Mozley, 2002).

$$\% = ([\text{O}_2]/[\text{O}_{\text{sat}}]) \times 100 \quad (2.6)$$

Dimana : % = persen saturasi oksigen

$[\text{O}_2]$ = konsentrasi oksigen yang diamati, dan

$[\text{O}_{\text{sat}}]$ = konsentrasi oksigen jenuh pada suhu tertentu

Konsentrasi oksigen jenuh pada temperatur tertentu dihitung dengan menggunakan persamaan dibawah ini (Mozley, 2002).

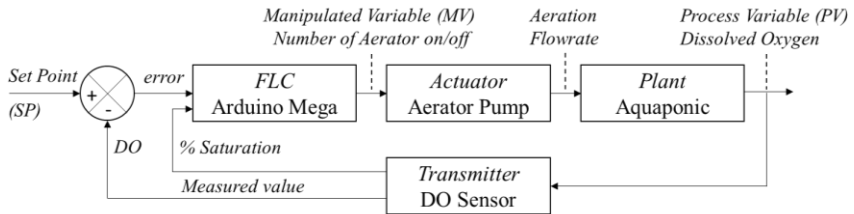
$$100\% \text{ Sat.O}_2 = 14.59 - 0.3955T + 0.0072T^2 - 0.0000619T^3 \quad (2.7)$$

Dimana : T = Temperatur air (°C).

2.2.3. Sistem Pengendali DO

Untuk bisa menjaga kadar oksigen terlarut pada *plant* akuaponik supaya pertumbuhan ikan dan tanaman bisa maksimal, diperlukan sebuah sistem kontrol yang handal. Sebuah sistem kontrol *closed loop feedback* digunakan untuk bisa melakukan eliminasi kesalahan pada plan dan mengatasi adanya *disturbance* atau gangguan (Ogata, 2010). Cara kerja sistem kontrol umpan balik adalah dengan terus memberikan nilai terukur dari variabel yang ingin dikendalikan kepada *controller*, sehingga apabila ada kesalahan akan terus dikoreksi supaya kesalahan tersebut hilang.

Sistem kontrol DO mulai bekerja dengan melakukan pengukuran (measured value) dari variabel proses (PV) yang ingin dikendalikan yaitu oksigen terlarut dengan menggunakan sensor DO. Nilai pengukuran akan dibandingkan dengan setpoint (SP) yaitu nilai DO yang diinginkan, sehingga akan diketahui nilai *error* atau kesalahannya. Kesalahan akan masuk ke *controller* sebagai *input* dan kemudian diproses untuk menghasilkan sinyal perintah berupa variabel manipulasi (MV) ke aktuator yaitu pompa aerasi. MV berupa berapa banyak pompa yang harus dinyalakan, dimana keputusan tersebut diambil dengan menggunakan algoritma *fuzzy logic controller*. Berapa banyaknya pompa aerator yang menyala berarti berapa banyak laju aliran oksigen (flowrate oksigen) yang diinjeksikan ke air. Dengan demikian kadar oksigen terlarut akan terus terkoreksi dan terjaga pada nilai *setpoint* yang diinginkan.



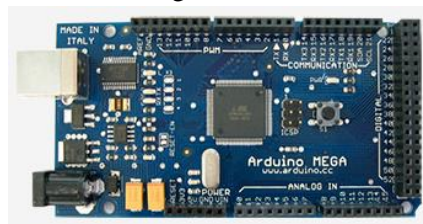
Gambar 2.8 Diagram Blok Sistem Pengendalian Kadar DO
(Cahya, 2016)

2.3. Komponen Sistem Pengendali

Sesuai dengan diagram blok pengendalian kadar oksigen terlarut, dibutuhkan komponen berupa, mikrokontroler, sensor, pompa aerator, serta komponen pendukung seperti catu daya, *relay*, dan rangkaian pengkondisian sinyal.

2.3.1. Mikrokontroler

Mikrokontroler merupakan *single chip* yang terdapat pada komputer atau berada pada divais kecil mikrokontroler yang digunakan untuk melakukan aksi pada sebuah objek, proses atau perintah. Mikrokontroler disebut sebagai *single chip* dikarenakan pada mikrokontroler mempunyai memori dan perangkat I/O yang berada di dalamnya (Axelson, 1997). Mikrokontroler yang digunakan adalah arduino Mega.



Gambar 2.9 Arduino Mega (Arduino, 2013)

Arduino mega adalah sebuah mikrokontroler berbasis Atmega2560. Arduino mega memiliki pin *input/output* sebanyak 54 (14 diantaranya bisa digunakan sebagai *output* PWM), pin input analog sebanyak 16, pin UARTs 4, sebuah *crystal oscillator* 16

MHz, koneksi USB, *power jack*, *ICSP header*, dan tombol reset. Untuk menghidupkan sistem, arduino mega hanya memerlukan catu daya yang bisa diambil dari USB komputer atau adaptor AC/DC atau menggunakan baterai 9V.

Tabel 2.2 Spesifikasi Arduino Mega

<i>Microcontroller</i>	Atmega2560
<i>Operating Voltage</i>	5V
<i>Input Voltage (recommended)</i>	7-12V
<i>Input Voltage (limits)</i>	6-20V
<i>Digital I/O Pins</i>	54 (15 bisa untuk keluaran PWM)
<i>Analog Input Pins</i>	16
<i>DC Current per I/O Pin</i>	20 mA
<i>DC Current for 3.3V Pin</i>	50 mA
<i>Flash Memory</i>	256 KB, 8 KB untuk <i>bootloader</i>
<i>SRAM</i>	8 KB
<i>EEPROM</i>	4 KB
<i>Clock Speed</i>	16 MHz

2.3.2. Sensor DO

Sensor DO adalah alat untuk mengukur kadar oksigen dalam air. Salah satu sensor DO tipe galvanik yang ada adalah sensor DO Atlas Scientific. Pada sensor DO tersebut, terdapat dua elektrode utama yang masing-masing berfungsi sebagai katode dan anode seperti sudah dijelaskan di bab 2.2.1. Spesifikasi sensor DO sebagai berikut.

Tabel 2.3 Spesifikasi Sensor DO

No	Spesifikasi	Nilai	Satuan
1	<i>Range</i>	0-35 +	mg/L
2	<i>Respon Time</i>	~ 0.3	mg/L/sec
3	<i>Max Pressure</i>	100	PSI

4	<i>Temperature Range</i>	1-50	°C
5	<i>Time before recalibration</i>	~ 1	Tahun
6	<i>Lifetime</i>	5 +	Tahun
7	<i>Maintenance</i>	~18	Bulan



Gambar 2.10 Sensor DO (AtlasScientific, 2015)

2.3.3. Rangkaian Pengkodisian Sinyal

Rangkaian pengkondisian sinyal adalah sebuah rangkaian yang digunakan untuk mengubah sinyal keluaran dari sensor supaya mampu dibaca oleh mikrokontroler. Sinyal dari sensor yang masih dalam skala mV, dikuatkan menjadi V dan atau dirubah menjadi format data tertentu yang kemudian diproses oleh mikrokontroler. Rangkaian pengkondisian sinyal sensor DO yang dipakai adalah dissolved oxygen EZO™ circuit. Spesifikasinya sebagai berikut.



Gambar 2.11 Rangkaian Pengkondisi Sinyal Sensor DO (AtlasScientific, 2015)

Tabel 2.4 Spesifikasi Rangkaian Pengkondisi Sinyal (AtlasScientific, 2015)

<i>Range</i>	<i>0.01 - 35.99 mg/L</i>
	<i>0.1 - 100 % saturation</i>
<i>Accuracy</i>	<i>+/- 0.05 mg/L</i>
<i>Max rate</i>	<i>1 reading/s</i>

<i>Supported probes</i>	<i>Any galvanic probe</i>
<i>Calibration</i>	<i>1 or 2 point</i>
<i>Temperature, salinity, and pressure compensation</i>	<i>Yes</i>
<i>Data protocol</i>	<i>UART & I₂C</i>
<i>Operating voltage</i>	<i>3.3V - 5V</i>
<i>Data Format</i>	<i>ASCII</i>

2.3.4. Pompa Aerator

Pompa aerator adalah sebuah alat yang digunakan untuk menambah kadar oksigen dalam air atau disebut aerasi. Aerasi adalah suatu proses penambahan udara/oksigen dalam air dengan membawa air dan udara ke dalam kontak yang dekat, dengan cara menyemprotkan udara ke dalam air atau dengan memberikan gelembung-gelembung halus udara dan membiarkannya naik melalui air (udara ke dalam air). Dengan cara demikian udara bebas (termasuk oksigen di dalamnya) yang diambil oleh pompa akan terdifusi dan terlarut kedalam air.



Gambar 2.12 Pompa Aerator TJ-2600 (Aquacave, 2017)

Pompa aerator yang akan digunakan adalah Turbo-Jet tipe TJ-2600 dengan *flowrate* maksimal sebesar 2100 L/jam dengan *head* sebesar 200 cm dan daya 32 W.

2.3.5. Relay

Relay adalah komponen listrik yang bekerja berdasarkan prinsip induksi medan elektromagnetis. Jika sebuah penghantar dialiri oleh arus listrik, maka di sekitar penghantar tersebut timbul medan magnet. Medan magnet yang dihasilkan oleh arus listrik tersebut selanjutnya diinduksikan ke logam ferromagnetis. *Relay* digunakan untuk memutus dan menghubungkan arus AC ke pompa aerator berdasarkan sinyal yang diberikan oleh mikrokontroller. Sinyal dari mikrokontroller tidak bisa langsung dihubungkan ke pompa aerator karena jenis tegangannya DC dan arusnya kecil.



Gambar 2.13 Relay (Omron, 2015)

2.3.6. Catu Daya

Adaptor adalah sebuah rangkaian yang berfungsi untuk mengubah tegangan AC menjadi tegangan DC. Adaptor yang digunakan adalah jenis adaptor *power supply* yang mengubah tegangan 220 volt AC menjadi tegangan 5 volt DC, sesuai dengan kebutuhan sensor, rangkaian pengkondisian sinyal, relay, dan mikrokontroler. Sedangkan untuk pompa aerator menggunakan tegangan AC yang dikontrol melalui relay.



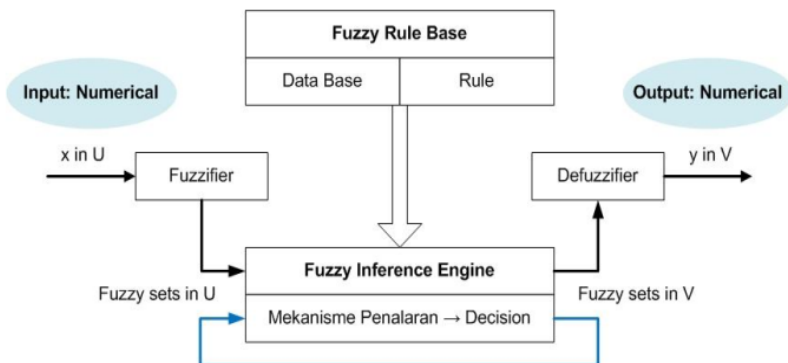
Gambar 2.14 Catu Daya (Tachopedia, 2017)

2.4. Logika Fuzzy

Logika *fuzzy* merupakan suatu metode untuk menentukan keputusan berbasis sebab dan akibat (if-then). Penggunaan logika *fuzzy* tidak memerlukan model matematis untuk mengetahui karakteristik sistem. Adapun kelebihan yang dimiliki oleh logika *fuzzy* sehingga logika *fuzzy* diterapkan pada dunia ilmu pengetahuan yaitu,

- Konsep matematis yang mendasari logika *fuzzy* mudah dimengerti.
- Logika *fuzzy* bekerja didasarkan bahasa linguistik dan mudah dipahami.
- Logika *fuzzy* mampu memodelkan fungsi-fungsi non linier yang kompleks.

Logika *fuzzy* terdiri dari 2 metode yaitu metode Mamdani, dan metode Takagi dan Sugeno. Penelitian ini menggunakan logika *fuzzy* dengan metode Mamdani. Perbedaan antara mamdani dan takagi-sugeno adalah jika *output fuzzy* mamdani berupa bilangan numerik, untuk *output fuzzy* takagi sugeno berupa sebuah fungsi. Adapun beberapa bagian penting dari logika *fuzzy* yaitu fuzzifikasi (*fuzzifier*), *fuzzy rule base*, *fuzzy inference engine*, dan defuzzifikasi (*defuzzifier*). Arsitektur logika *fuzzy* Mamdani ditunjukkan oleh Gambar 2.15.



Gambar 2.15 Arsitektur logika *fuzzy* Mamdani (Wang, 1997)

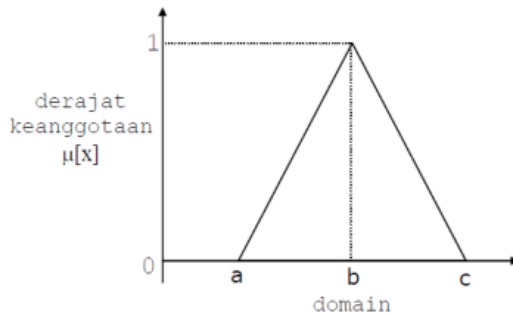
Metode Mamdani sering juga dikenal dengan nama Metode MaxMin. Metode ini diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Metode Mamdani terdiri dari 3 tahapan (Wang, 1997) yaitu,

a) Fuzzifikasi (*Fuzzifier*)

Masukan dan keluaran sistem dalam logika *fuzzy* akan dinyatakan dalam fungsi keanggotaan. Fungsi keanggotaan merupakan suatu *range* nilai yang dapat merepresentasikan bentuk masukan atau keluaran sistem. Pada proses fuzzifikasi, masukan *fuzzy* akan diubah menjadi linguistik dan diolah dalam *fuzzy inference engine*. *Input* berupa numerik di petakan menjadi linguistik berdasarkan fungsi keanggotaan. Fungsi keanggotaan himpunan *fuzzy* adalah sebuah kurva untuk memetakan masukan menjadi derajat keanggotaan *fuzzy*. Ada beberapa representasi kurva beserta persamaannya yaitu:

- **Representasi Kurva Segitiga**

Kurva segitiga pada dasarnya merupakan gabungan antara dua garis (linier) seperti pada Gambar 2.16



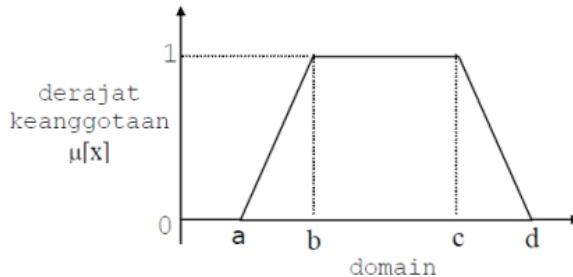
Gambar 2.16 Fungsi Keanggotaan Segitiga

Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ (x - a)/(b - a); & a \leq x \leq b \\ (b - x)/(c - b); & b \leq x \leq c \end{cases}$$

- **Representasi Kurva Trapesium**

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan 1 seperti Gambar 2.17.



Gambar 2.17 Fungsi Keanggotaan Trapesium

Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \leq a \text{ atau } x \geq d \\ (x-a)/(b-a); & a \leq x \leq b \\ 1; & b \leq x \leq c \\ (d-x)/(d-c); & x \geq d \end{cases}$$

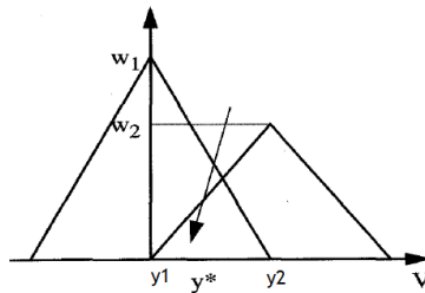
b) Fuzzy Inference Engine

Masukan sistem yang telah di-fuzzifikasi menjadi himpunan *fuzzy* akan diolah dalam sebuah *fuzzy inference engine*. Di dalam *fuzzy inference engine* terdapat *fuzzy rule base*. *Rule base* adalah kumpulan aturan yang akan digunakan oleh FIS dalam mengambil (implikasi) kesimpulan berdasarkan masukan yang diberikan. Setiap masukan akan dihubungkan dengan operator matematis seperti *and*, *or*, dan *not*. Masukan yang sudah dipetakan dalam himpunan *fuzzy* akan diolah oleh *fuzzy inference engine* untuk menghasilkan keluaran *fuzzy* dengan komposisi if-then yang ada di dalam *fuzzy rule base*. Di dalam *fuzzy inference engine* juga terdapat fungsi agregasi. Bila fungsi implikasi digunakan untuk menentukan bentuk akhir *fuzzy* set keluaran berdasarkan tiap-tiap *rule-base*, maka selanjutnya dilakukan fungsi agregasi yaitu proses mengkombinasikan seluruh keluaran (if-then) menjadi sebuah *fuzzy* tunggal. Fungsi implikasi yang digunakan adalah Min dan fungsi agregasi yang digunakan adalah Max.

c) Defuzzifikasi (*Defuzzifier*)

Keluaran dari *fuzzy inference engine* masih berupa besaran linguistik. Dengan proses defuzzifikasi, besaran linguistik tersebut akan diubah menjadi numerik sesuai dengan fungsi keanggotaan keluaran sistem *fuzzy* tersebut. Angka numerik yang dihasilkan merupakan keluaran dari logika *fuzzy*. Ada beberapa metode defuzzifikasi seperti *largest of maximum* (LOM), *middle of maximum* (MOM) dan *center of area* (COA) atau centroid =. Metode defuzzifikasi yang digunakan pada penelitian ini adalah metode *centroid*. Pada metode centroid solusi crisp diperoleh dengan menghitung titik beban (pusat gravitasi) dengan persamaan berikut,

$$y^* = \frac{y_1 w_1 + y_2 w_2}{w_1 + w_2} \quad (2.8)$$



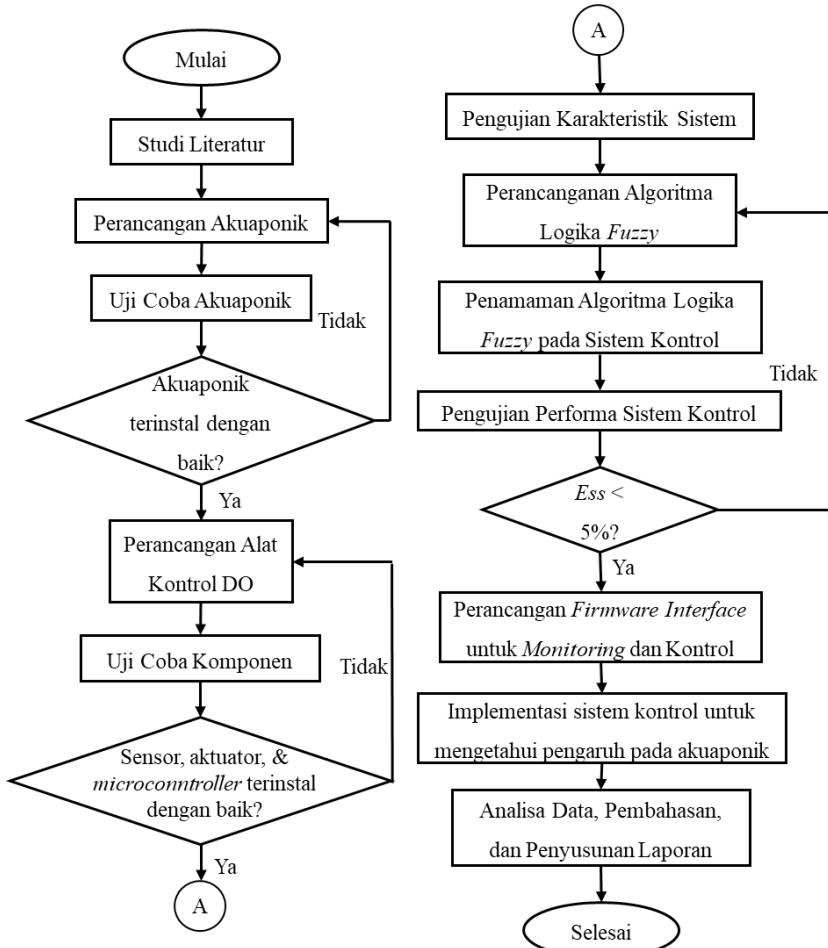
Gambar 2.18 Representasi Kurva *Centroid*

Representasi kurva dengan metode *centroid* dapat dilihat pada Gambar 2.18 yang ditunjukkan oleh anak panah pada dengan simbol. Metode *centroid* sering digunakan pada tahapan defuzzifikasi di dalam sistem *fuzzy* (Wang, 1997).

BAB III METODOLOGI PENELITIAN

3.1. Alur Penelitian

Secara garis besar, alur dari penelitian yang dilakukan tercantum pada diagram alir berikut:



Gambar 3.1 Diagram Alir Tahapan Penelitian

Penelitian dilakukan dengan melakukan studi literatur terlebih dahulu, yaitu mencari dasar permasalahan dan dasar pemecahan masalahnya berdasarkan jurnal atau penelitian sebelumnya. Setelah itu, dibuat *plant* akuaponik yang akan dijadikan objek eksperimen. Akuaponik tentu harus diuji sirkulasi airnya apakah sudah terinstal dengan baik dan tidak ada masalah seperti kebocoran. Perancangan alat sistem pengendalian kadar oksigen terlarut (mekanik, elektrik, dan program) dibuat setelahnya dan dilakukan pengujian kemudian. Program awal yang dibuat masih sebatas pada pembacaan nilai DO dari sensor dan mengontrol aerator sebatas *on/off*.

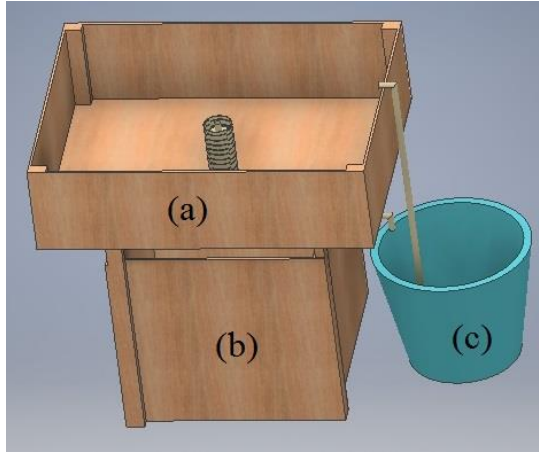
Sebelum algoritma sistem kontrol dibuat, pengujian karakteristik pompa aerator dilakukan agar penentuan MF *input*, MF *output*, dan *rule base* pada FLC bisa didefinisikan dengan benar. Pendefinisian yang salah akan menyebabkan performansi kontrol yang buruk. Performa alat kemudian diuji. Apabila masih kesalahan (ess) masih diatas batas yang ditentukan yaitu 5%, perbaikan algoritma kontrol harus dilakukan, diimplementasikan, dan diuji kembali sampai ess dibawah 5%. Setelah FLC bekerja dengan baik maka dilakukan perancangan firmware untuk *interface monitoring* dan kontrol pada alat. *Firmware* dibuat dengan mengkonfigurasi alat agar bisa berkomunikasi dengan *platform* yang dipakai. *Platform* kemudian akan seting agar mampu melakukan pembacaan data yang dikirim oleh alat. Setelah itu maka tahap yang terakhir sebelum menganalisa adalah mengimplementasikan sistem kontrol kadar oksigen terlarut pada akuaponik untuk mengetahui apakah pengendalian kadar oksigen terlarut bisa memberikan pengaruh pada ikan dan tanaman akuaponik.

3.2. Studi Literatur

Berdasarkan hasil studi referensi, setpoint kadar oksigen untuk akuaponik yang baik adalah 5.5 mg/L. Sehingga pada *controller* yang dirancang akan digunakan *setpoint* sebesar 5.5 mg/L. Untuk pengujian *controller* toleransi *error* yang diijinkan adalah 5% dari *setpoint* atau 0,25 mg/L. Tingkat keberhasilan alat juga akan diukur dengan melihat pengaruh sistem kontrol pada tingkat pertumbuhan ikan dan tanaman yang akan dilakukan pengamatan selama 10 hari.

3.3. Perancangan Akuaponik

Kolam akuaponik dan media tanam didesain dengan ukuran kecil. Hal ini untuk mempermudah uji coba alat dan pengamatan pengaruhnya ke ikan dan tanaman.



Gambar 3.2 Desain Kolam Akuaponik

Desain akuaponik terbagi menjadi dua bagian, yaitu kolam untuk media pemeliharaan ikan dibagian bawah (b) dan tempat media tanam untuk tanaman (a) serta satu ember tambahan sebagai *sump tank* (c) yang berfungsi supaya air pada kolam ikan tidak naik turun ketika sirkulasi air dijalankan. Ukuran kolam ikan adalah 63 x 63 x 50 cm atau setara dengan 225 liter air. Dimensi media tanam adalah 105 x 70 x 16 cm atau setara dengan 110 liter, dimana 30% atau 35 liter adalah air dan 75 liter merupakan media tanam berupa pasir malang (pasir hitam dengan butiran yang besar). Pasir malang dipilih karena memiliki *specific surface area* yang besar yaitu 280 m^2/m^3 yang bagus untuk perkembangan bakteri pengurai, namun ukurannya cukup besar sehingga air bisa mengalir dengan lancar. *Sump tank* yang digunakan memiliki volume 40 liter. Akuaponik akan diisi dengan 32 ikan nila dan 18 sayur selada.

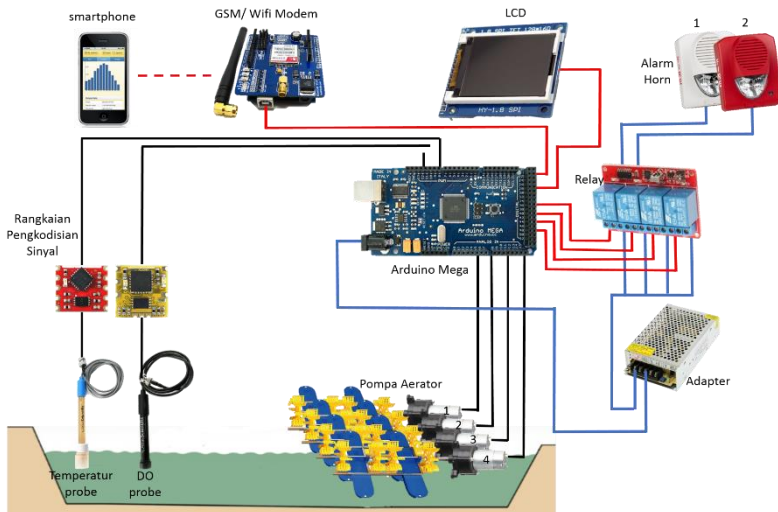
Sistem akuaponik yang dibuat bekerja seperti yang sudah dijelaskan pada bab 2.1. Sistem sirkulasi air pada akuaponik menggunakan metode *ebb and flow* atau pasang surut, yaitu air akan

dipompa dari *sump tank* menuju ke media tanam. Setelah media tanam terisi penuh oleh air pada batas yang ditentukan, *bell siphon* yang ada di tengah akan menghisap air pada media tanam hingga air pada media tanam hampir kosong. *Bell siphon* menggunakan bekerja menggunakan hukum pascal. Kemudian pengisian akan terjadi lagi dan terus berulang pasang surut.

Penempatan pompa aerator diletakkan pada keempat sisi kolam pada dasar kolam. Hal ini agar persebaran dari kadar oksigen terlarut. Penempatan dari kadar sensor DO diletakkan pada titik tengah kolam ikan dengan ketinggian 25 cm (ditengah) karena yang paling mewakili nilai DO keseluruhan dari kolam.

3.4. Perancangan Sistem Kontrol DO

Perancangan sistem kontrol DO dibuat dengan membuat skema rangkaian *hardware* atau komponen-komponen elektrik dan mekanik sesuai dengan diagram blok pada gambar 2.7.



Gambar 3.3 Skematik Rangkaian Alat Pengendali Kadar DO

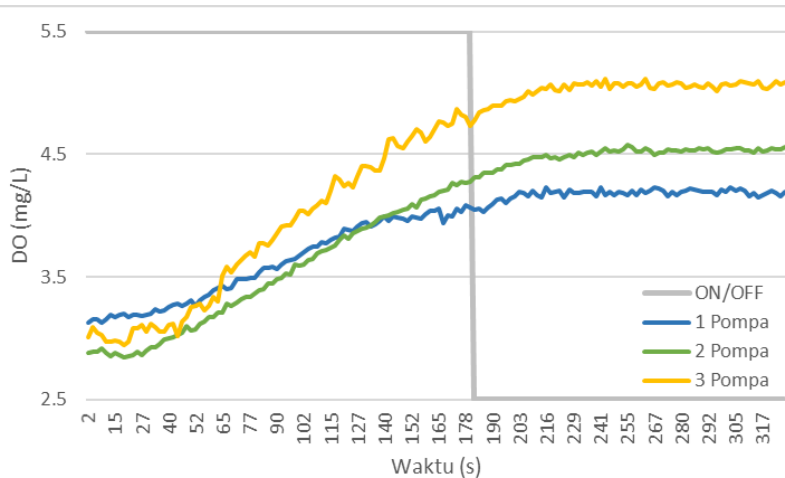
Seperti terlihat pada gambar 3.3, sistem akan mulai bekerja dengan mengukur kadar DO pada kolam menggunakan sensor DO.

Variabel temperatur juga diukur dengan sensor *thermocouple*. Temperatur diukur karena mempengaruhi nilai % saturasi dari DO. Pada nilai % saturasi yang berbeda, kemampuan air untuk menangkap oksigen juga berbeda. (Atlas Scientific, 2016) Semakin tinggi % saturasi semakin sulit oksigen terdifusi dalam air. Pada temperatur yang tinggi, aktivitas ikan lebih tinggi sehingga mengkonsumsi oksigen lebih banyak.

Masing-masing data pembacaan sensor kemudian masuk ke rangkaian pengkondisian sinyal supaya *output* sensor berupa arus dan tegangan bisa di konversi dan dibaca oleh mikrokontroler. Pada mikrokontroler sudah ditanam algoritma kontrol FLC yang akan melakukan manipulasi pompa aerator berdasarkan *error* yang masuk dan % saturasi DO. Pompa aerator yang disuplai tengangan AC diberi perantara berupa *relay* karena output dari mikrokontroler berupa sinyal DC 5 volt 50 mA. Sinyal DC tersebut yang akan mengatur *on/off relay* secara otomatis untuk memutus dan menghubungkan arus AC dari sumber ke pompa aerator. Seluruh data *monitoring* DO dan temperatur serta status kondisi pompa aerator akan ditampilkan pada layar LCD dan juga dikirimkan ke server melalui jaringan internet menggunakan modul *wifi* yang terhubung ke mikrokontroler. Data tersebut bisa diakses oleh pengguna melalui web atau dengan membuka aplikasi *smartphone*. Seluruh daya yang dibutuhkan oleh komponen elektrik di suplai oleh catu daya atau *adaptor* 5 dan 12 VDC. Alat juga dilengkapi dengan alarm yang akan mengingatkan pengguna apabila kualitas air di luar batas yang ditentukan dan sistem mengalami kegagalan.

3.5. Pengujian Karakteristik Sistem

Pengujian karakteristik sistem dilakukan untuk mengetahui bagaimana perubahan pada sistem akan berubah terhadap perlakuan yang diberikan. Dengan mengetahui karakteristik dari sistem maka sistem bisa dikendalikan. Percobaan dilakukan dengan memanipulasi pompa aerator dengan berbagai kondisi. Grafik perubahan DO pada akuaponik yang dipengaruhi oleh pompa ditunjukkan gambar 3.5.

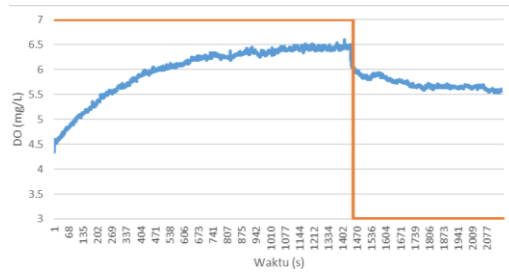


Gambar 3.4 Karakteristik Respon DO Akibat Pompa Aerator

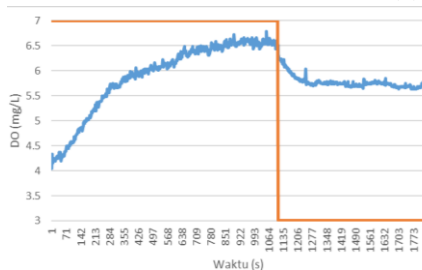
Dari percobaan pengujian karakteristik pompa, didapatkan hasil seperti terlihat pada Gambar 3.5. Pengujian open loop dengan menyalakan pompa selama 3 menit menunjukkan bahwa semakin banyak jumlah pompa semakin tinggi kenaikan kadar oksigen terlarutnya. Untuk 1 pompa aerator naik dari 3.13 mg/L menjadi 4.19 mg/L atau naik 1.06 mg/L. Untuk 2 pompa aerator naik dari 2.85 mg/L menjadi 4.44 mg/L atau naik 1.59 mg/L. Untuk pompa 1 naik dari 3.01 mg/L menjadi 5.09 mg/L atau naik 2.08 mg/L. Setiap penambahan 1 pompa dalam waktu 3 menit terjadi kenaikan 0.49 – 0.53 mg/L. Dua puluh detik setelah pompa aerator dimatikan bisa diamati bahwa kenaikan DO masih terjadi. Hal ini menunjukkan bahwa oksigen yang terdifusi dalam air membutuhkan waktu untuk untuk terdifusi secara menyeluruh.

Pengujian karakteristik sistem dengan menyalakan pompa sampai mencapai kondisi *steady* (saat kondisi jenuh) dilakukan dengan hasil seperti pada Gambar 3.6. Semakin banyak pompa menunjukkan sistem semakin cepat mencapai *steady state* atau jenuh. Namun, pada saat pompa dimatikan, kadar oksigen langsung mengalami penurunan dari ± 6.6 mg/L menjadi $\pm 5,7$ mg/L. Hal ini dikarenakan kadar oksigen dalam air sudah melebihi nilai

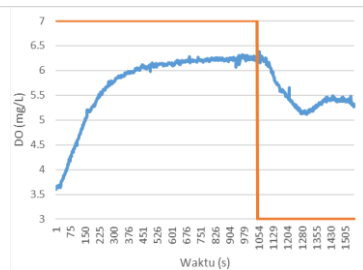
saturasinya sehingga terlepas ke udara bebas. Saat DO mencapai 6.6 mg/L nilai saturasinya 89% dan saat 5.7 mg/L nilai saturasinya adalah 75%. Artinya ketika persen saturasi dari DO sudah diatas 75%, air sudah tidak mampu untuk menyerap lebih banyak oksigen dari udara bebas. Berdasarkan hasil uji, diketahui bahwa respon DO tidak bersifat linier melainkan bersifat eksponensial yaitu semakin tinggi % saturasi semakin kecil kenaikan DO yang diperoleh dengan waktu yang sama.



(a)



(b)



(c)

Gambar 3.5 Respon DO Ketika Dinyalakan Pompa Aerator Sebanyak (a) 1 Pompa, (b) 2 Pompa, dan (c) 3 Pompa Sampai Mencapai Keadaan Jenuh.

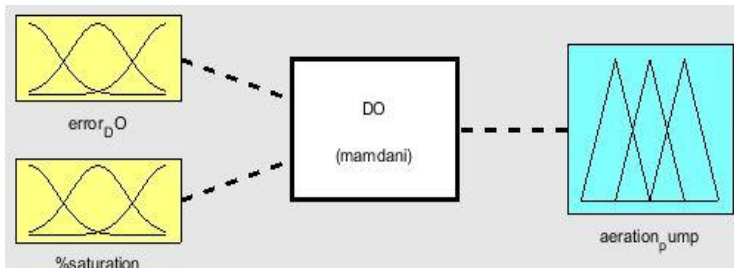
3.6. Perancangan Algoritma Logika Fuzzy

Berdasarkan karakteristik yang sudah didapatkan, langkah berikutnya adalah membuat algoritma kontrol dengan logika *fuzzy*. Perancangan ini berupa pembuatan *fuzzy inference system* (FIS) pada software Matlab. Penelitian ini menggunakan logika *fuzzy* dengan metode Mamdani karena membutuhkan kesimpulan (consequent

atau crisp) dan fungsi keanggotaan keluaran berupa *fuzzy sets* yang memiliki nilai pada tiap fungsi keanggotaannya. Terdapat 3 tahapan perancangan logika *fuzzy* yaitu fuzzifikasi, *fuzzy inference engine*, dan defuzzifikasi.

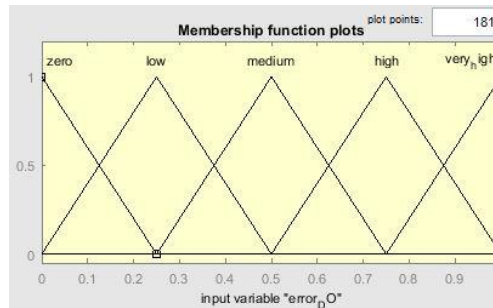
3.7.1 Fuzzifikasi

Fuzzy inference system untuk sistem pengendali kadar oksigen terlarut terdiri dari dua masukan dan satu keluaran. Variabel yang menjadi masukan FIS adalah *error DO* (selisih antara nilai DO terukur dengan *set point*) dan % saturasi (berapa nilai kadar DO terukur dibandingkan nilai jenuhnya pada suhu tertentu). Dari kedua variabel tersebut, FIS akan menghasilkan sebuah keluaran berupa jumlah pompa aerator yang dihidupkan untuk meningkatkan kadar oksigen. Diagram FIS sistem pengendali DO ditunjukkan oleh Gambar 3.6



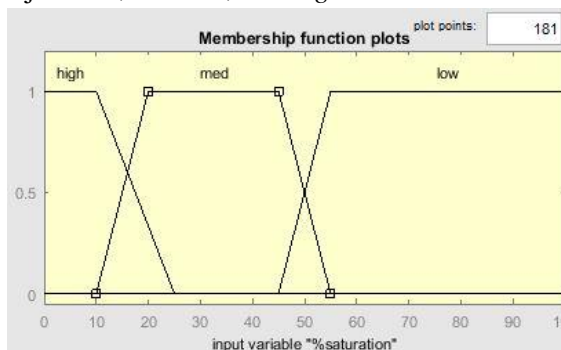
Gambar 3.6 FIS Sistem Pengendali DO

Pada masukan *error DO*, fungsi keanggotaan terbagi menjadi lima yaitu *zero*, *low*, *medium*, *high*, dan *very high error*. Fungsi keanggotaan tersebut berdasarkan besarnya *error* pada *plant* akuaponik yaitu berturut-turut 0, 0.25, 0.5, 0.75, 1.0 mg/L, seperti yang ditunjukkan pada Gambar 3.7. Nilai *error* tersebut dipilih berdasarkan karakteristik dari pompa dimana jumlah pompa sebanding dengan kenaikan DO yang diperoleh.



Gambar 3.7 Fungsi Keanggotaan *error DO*

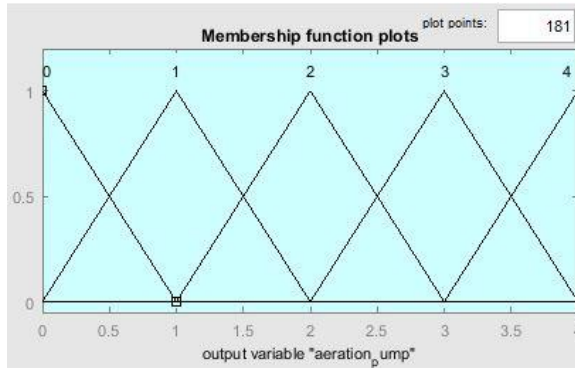
Masukkan yang kedua berupa % saturasi merupakan nilai yang menunjukkan tingkat kejenuhan oksigen di dalam air. %saturasi dihitung dengan menggunakan persamaan 2.6 dan 2.7. Kadar oksigen terlarut memiliki model yang eksponensial yang berarti semakin mendekati 100% jenuh, maka semakin kecil kenaikannya. Hasil uji karakteristik menunjukkan bahwa semakin besar % saturasi DO, semakin lama kenaikan DO terhadap waktu. Sehingga, tingkat saturasi yang berbeda harus menjadi pertimbangan FIS dalam memberikan nilai *output* yang sesuai. Fungsi keanggotaan % saturasi terbagi menjadi *low*, *medium*, dan *high*.



Gambar 3.8 Fungsi Keanggotaan % Saturasi

Berdasarkan kedua masukan tersebut, FIS akan menentukan *ouput* berupa jumlah pompa aerator yang harus dinyalakan pada saat tersebut untuk menjaga kadar oksigen dalam air. Fungsi

keanggotaan *output* terbagi menjadi 0, 1, 2, 3, dan 4 pompa aerator. Nilai *output* menunjukkan banyaknya aerator yang menyala. *Output* 0 berarti tidak ada pompa aerator yang menyala. *Output* 1 berarti menunjukkan 1 pompa aerator menyala untuk mensuplai oksigen dan seterusnya.



Gambar 3.9 Fungsi Keanggotaan Pompa Aerator

3.7.2 Fuzzy Inference Engine

Di dalam *fuzzy inference engine* terdapat aturan logika *fuzzy* (fuzzy rule base) berupa perintah untuk menentukan keluaran berdasarkan nilai masukannya. Kedua *input* dihubungkan dengan operator matematis *and* dan *or* dengan logika penarikan kesimpulan (implikasi) *if-then* menjadi sebuah *rule base*. Penelitian ini menggunakan sembilan aturan logika *fuzzy* dapat dilihat pada Gambar 3.10. Hasil dari keseluruhan keluaran *rule-base* di agregasi atau dikombinasi menjadi *fuzzy set* tunggal yang akan menentukan kerja aktuator yaitu pompa aerator seperti terlihat pada tabel 3.1.

Tabel 3.1 Hubungan *Input-Output* berdasarkan *Rule Base*.

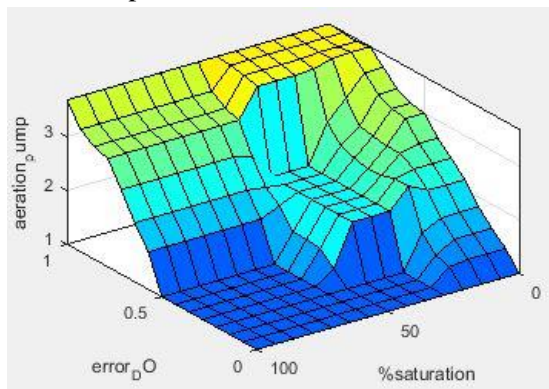
Error DO % Sat	Zero	Low	Medium	High	Very High
Low	0	0	1	2	3
Medium	0	1	2	3	4
High	1	2	3	4	4

1. If (error_DO is zero) then (aeration_pump is 1) (1)
2. If (error_DO is low) and (%saturation is low) then (aeration_pump is 1) (1)
3. If (error_DO is low) and (%saturation is high) then (aeration_pump is 2) (1)
4. If (error_DO is high) and (%saturation is low) then (aeration_pump is 3) (1)
5. If (error_DO is high) and (%saturation is high) then (aeration_pump is 4) (1)
6. If (error_DO is very_high) then (aeration_pump is 4) (1)
7. If (error_DO is medium) and (%saturation is med) then (aeration_pump is 2) (1)
8. If (error_DO is medium) and (%saturation is high) then (aeration_pump is 3) (1)
9. If (error_DO is medium) and (%saturation is low) then (aeration_pump is 1) (1)

Gambar 3.10 Aturan Logika Fuzzy

3.7.3 Defuzzifikasi

Pada proses defuzzifikasi, besaran linguistik dari hasil agregasi akan diubah menjadi numerik sesuai berdasarkan fungsi keanggotaan keluaran sistem *fuzzy* tersebut. Angka numerik yang dihasilkan merupakan keluaran dari logika *fuzzy*. Metode defuzzifikasi yang digunakan pada penelitian ini adalah metode *centroid*. Grafik pemetaan *output* terhadap *input* pada proses defuzzifikasi terlihat pada Gambar 3.11



Gambar 3.11 Pemetaan *Input Output* FIS

3.7. Pengujian Performa Sistem Kontrol

Pengujian performa sistem kontrol dilakukan dengan dua metode yaitu *tracking set point* dan uji *disturbance*. Pengujian ini dilakukan untuk mengetahui apakah respon DO dapat mencapai set point yang ditentukan dengan toleransi kesalahan $\pm 5\%$.

Pengujian *tracking set point* dilakukan dengan merubah *set point* DO mulai dari 3 mg/L menuju 4.5 mg/L menuju 5.5 mg/L menuju 6.0 mg/L. Pengujian *disturbance* dilakukan dengan memberikan gangguan pada air kolam berupa air dengan kadar *dissolved oxygen* rendah yaitu air limbah budidaya ikan dengan nilai $DO \pm 1$ mg/L, sebanyak 10 L, 20 L, dan 30 L. Air yang digunakan sebagai gangguan system ditunjukkan pada Gambar 3.12



Gambar 3.12 Air Limbah Budidaya sebagai Gangguan

3.8. Perancangan *Interface Monitoring* dan Kontrol

Interface untuk melakukan *monitoring* dan kontrol dibuat untuk memudahkan pengamatan dan perekaman data pada saat alat bekerja. *Interface* dibuat dengan menggunakan platform *internet of things* (IoT) thinger.io, dimana alat yang dibuat terkoneksi dengan jaringan internet sehingga bisa mengirimkan data dan menerima perintah secara *real time* dari *server*. *Monitoring* dan kontrol bisa dilakukan melalui web yang bisa dibuka di PC ataupun *smartphone*. Pembuatan *interface* dibuat dengan melakukan pengaturan jaringan pada program yang ditaman agar bisa berkomunikasi dengan *server*. Selain itu dilakukan pembuatan *layout* tampilan pada web untuk menampilkan data-data yang akan ditampilkan. Hasil dari *interface monitoring* dan kontrol terlihat pada Gambar 3.13



Gambar 3.13 *Interface Sistem Monitoring dan Kontrol DO*

3.9. Implementasi Sistem Kontrol

Setelah keseluruhan sistem kontrol sudah bekerja, alat kemudian diuji untuk diimplementasikan pada akuaponik. Pengujian dilakukan selama 10 hari untuk melihat pengaruh dari pengendalian DO terhadap laju pertumbuhan ikan dan tanaman pada akuaponik. Pada pengujian ini, digunakan dua akuaponik, satu kadar oksigennya dikontrol dan satunya tidak. Perbandingan dilakukan dengan mengukur berat ikan dan tinggi tanaman pada awal dan akhir waktu pengamatan. Perhitungan laju pertumbuhan dilakukan dengan persamaan 2.1.

Halaman ini memang dikosongkan

BAB IV

ANALISA DATA DAN PEMBAHASAN

4.1. Uji Karakteristik Alat

Pengujian karakteristik dari alat meliputi kalibrasi sensor DO dan uji pompa aerator. Pengujian dilakukan untuk mengetahui keakuratan dari pembacaan sensor DO yang digunakan, untuk mengetahui kemampuan pompa aerator memsuplei oksigen dalam air, dan karakteristik dari plan akuaponik yang dikendalikan. Beberapa uji karakteristik pompa aerator sudah dibahas pada bab 3.

4.1.1 Kalibrasi Sensor

Pengujian dilakukan dengan membanding hasil pembacaan sensor DO yang dipakai dengan alat ukur yang lebih standar. Alat ukur yang digunakan sebagai pembanding adalah YSI 550A dan dilakukan di Laboratorium Basah, Jurusan Perikanan, Universitas Airlangga.



Gambar 4.1 Kalibrasi Sensor DO dengan YSI 550A

Hasil kalibrasi terlihat seperti pada tabel 4.1. Terlihat bahwa hasil pembacaan sensor DO dengan YSI 550A sangat mendekati. Dari data tersebut, hanya pada data ke 5 yang memiliki *error* cukup besar yaitu 0,319 mg/L. Namun hal tersebut bisa diabaikan karena pembacaan DO pada kolam akuaponik sangat jarang mencapai nilai diatas 8 mg/L. Untuk *error* rata-rata didapatkan sebesar 0,104 dan

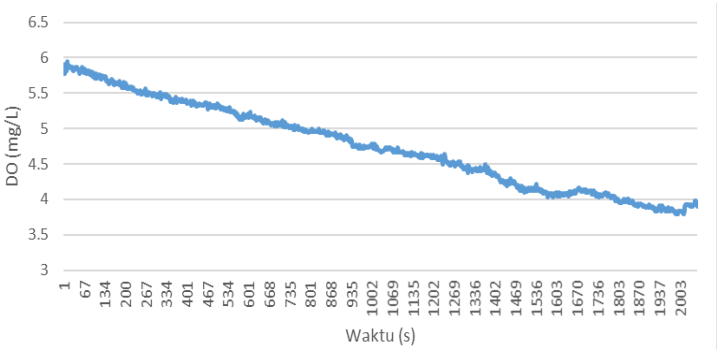
standar deviasi sebesar 0.072 ± 0.00931 . Sensor DO yang dipakai dapat dinyatakan akurat dan layak untuk digunakan.

Tabel 4.1 Hasil Kalibrasi Sensor DO

Pembacaan std	Pembacaan Alat					<i>Error</i>
	1	2	3	4	Rata-rata	
5.784	5.64	5.7	5.73	5.77	5.7248	0.0592
6.914	6.82	6.84	6.85	6.86	6.8568	0.0572
7.276	7.29	7.31	7.3	7.33	7.3012	-0.025
7.584	7.48	7.52	7.51	7.52	7.5228	0.0612
8.606	9	9.01	9	9.01	8.9252	-0.319

4.2. Uji *Open Loop* Sistem

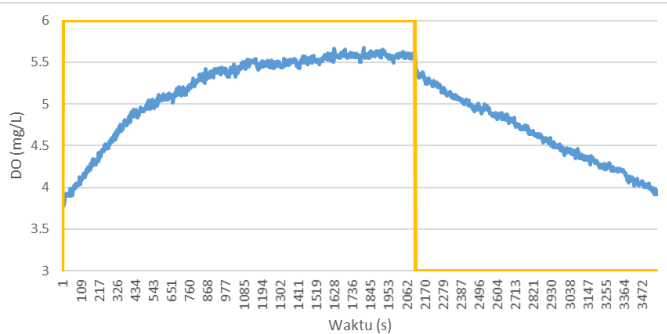
Pengujian *Open loop* dilakukan untuk mengetahui bagaimana karakteristik dari sistem yang akan dikendalikan. Hasil uji open loop pada akuaponik dapat dilihat pada Gambar 4.3.



Gambar 4.2 Perubahan DO disebabkan oleh ikan

Pengujian dilakukan dengan memberikan 40 ikan berumur 3 bulan kedalam wadah kolam yang terisi air sebanyak 80 L. Penurunan kadar oksigen dalam air kemudian diukur sebagai fungsi waktu. Dapat diamati bahwa penurunan terjadi mulai dari 5.9 mg/L menjadi 3.9 mg/L selama 2060 detik. Sehingga dapat dihitung satu ikan nila berumur 3 bulan memiliki laju konsumsi kadar oksigen sebesar 0.728×10^{-3} mg/L/menit untuk beraktifitas. Supaya

penurunan kadar oksigen terlarut tersebut bisa kembali naik, dibutuhkan aerator untuk mensuplai oksigen dalam air. Maka dilakukan uji *open loop* pompa aerator.



Gambar 4.3 Perubahan DO disebabkan oleh pompa

Setelah terjadi penurunan akibat ikan, maka diberi perlakuan berupa dinyalakan 1 pompa sampai mencapai *steady state* dan dimatikan kembali seperti terlihat pada Gambar 4.4. Dengan menggunakan 1 pompa. Kadar oksigen dalam air mampu naik dari 3.9 mg/L menjadi 5.5 mg/L selama 20 menit. Dan ketika pompa dimatikan DO akan turun kembali karena digunakan oleh ikan.

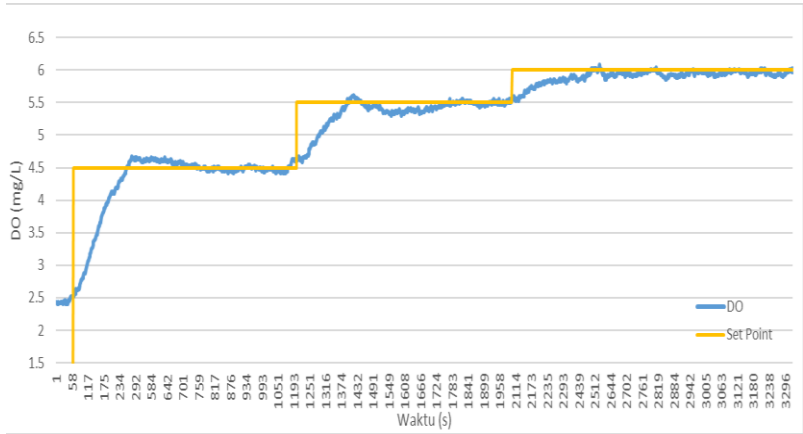
4.3. Uji *Closed Loop* Sistem

Pengujian ini dilakukan untuk mengetahui apakah respon DO dapat mencapai set point yang ditentukan dengan toleransi kesalahan $\pm 5\%$. Pengujian dilakukan dengan merubah *set point* DO (tracking setpoint) dan dengan memberikan *disturbance* pada air kolam berupa air dengan kadar *dissolved oxygen* rendah yaitu air limbah budidaya ikan dengan nilai DO ± 1 mg/L, sebanyak 10 L, 20 L, dan 30 L.

4.4.1 Uji *Closed Loop* Tracking Setpoint

Pengujian *tracking set point* dilakukan untuk mengetahui apakah sistem mampu bekerja pada *range* yang ditentukan. Pengujian ini juga bertujuan untuk mengetahui kemampuan

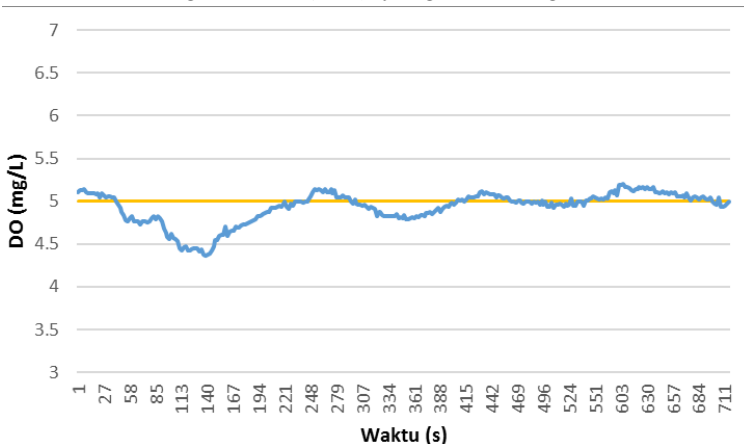
pengendali yang telah dirancang dalam mengatasi kondisi proses pada saat diberi perubahan *set point*.



karena nilai saturasinya yang sudah mencapai 89%. Keseluruhan *maximum overshoot* tidak mencapai 5 % yang berarti kerja sistem kontrol baik.

4.4.2 Uji *Closed Loop* dengan Gangguan 10 L Limbah Budidaya

Pengujian *disturbance* dilakukan dengan memberikan air dengan kadar *dissolved oxygen* rendah yaitu air limbah budidaya ikan dengan nilai $DO \pm 1$ mg/L. Hasil pengujian *closed loop* dengan gangguan 10 L air kolam dengan dapat dilihat pada Gambar 4.5. Nilai DO dikondisi awal adalah sebesar 5.1 mg/L. Dengan adanya gangguan air kolam yang menurunkan kadar oksigen, pompa aerator aktif sesuai dengan aturan *fuzzy* yang dirancang.



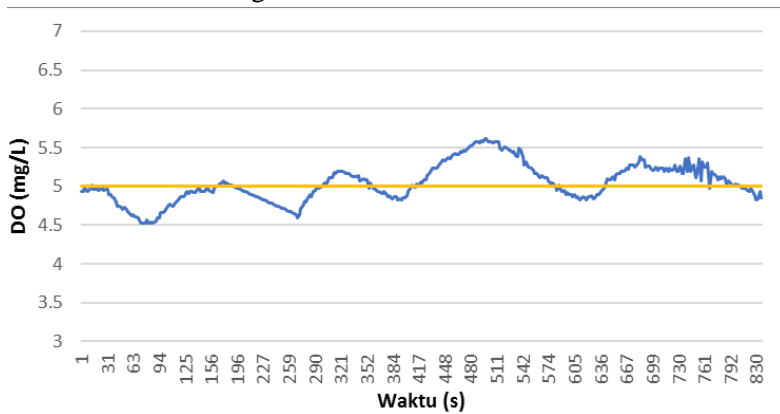
Gambar 4.5 Respon DO terhadap *disturbance* 10 L

Berdasarkan hasil uji, dapat dilihat bahwa DO yang turun sampai 4.4 mg/L mampu naik pada set point 5 mg/L kembali dengan adanya sedikit osilasi. Hal ini dikarenakan karakteristik oksigen yang memerlukan waktu untuk terdifusi. Di menit ke 8, oksigen sudah cukup stabil. Dengan demikian performa sistem kontrol cukup baik. Performa sistem kontrol sebagai berikut *settling time* (t_s) = 524 detik, *error steady state* (ess) = 0.12 mg/L, maksimum *overshoot* = 3.8%. Setelah mencapai *steady state*, rata-rata *error* DO adalah 0.11

mg/L atau 2.2%, dimana masih dalam toleransi pengendalian DO yaitu 5%

4.4.3 Uji *Closed Loop* dengan Gangguan 20 L Limbah Budaya

Hasil pengujian *closed loop* dengan gangguan 20 L air kolam dengan dapat dilihat pada Gambar 4.6. Nilai DO dikondisi awal adalah sebesar 4.9 mg/L.

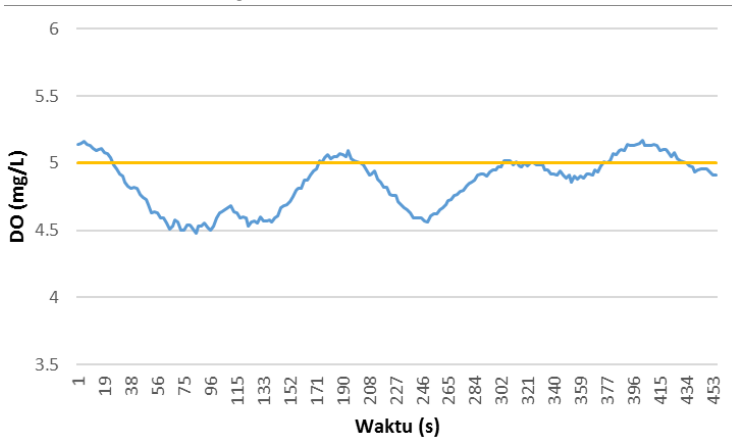


Gambar 4.6 Respon DO terhadap disturbance 20 L

Berdasarkan hasil uji, dapat dilihat bahwa DO yang turun sampai 4.5 mg/L kemudian naik mengikuti *set point* namun berkali kali berosilasi. Pada menit ke 8, terjadi *overshoot* yang cukup besar yaitu 11,6 %. Hal ini dikarenakan karakteristik oksigen yang memerlukan waktu yang cukup lama untuk terdifusi dari air yang mengandung banyak oksigen terlarut ke air yang kadar oksigen terlarutnya rendah. Respon sistem kurang stabil kemungkinan karena gangguan yang diberikan cukup besar yaitu seperempat dari kapasitas kolam sendiri. Dengan demikian performa sistem kontrol harus diperbaiki untuk *disturbance* yang sangat besar. Performa sistem kontrol melewati sempit melewati batas toleransi kesalahan yang diperbolehkan yaitu 5%. Didetik ke 730 terjadi keanehan pada grafik seperti terjadi *noise* pada pembacaan sensor. Pembacaan menunjukkan ketidakstabilan dan terjadi perubahan yang sangat cepat yang biasanya tidak pernah terjadi.

4.4.4 Disturbance 30 L Limbah Budidaya

Hasil pengujian closed loop dengan gangguan 30 L air kolam dengan dapat dilihat pada Gambar 4.7. Nilai DO dikondisi awal adalah sebesar 5.15 mg/L.



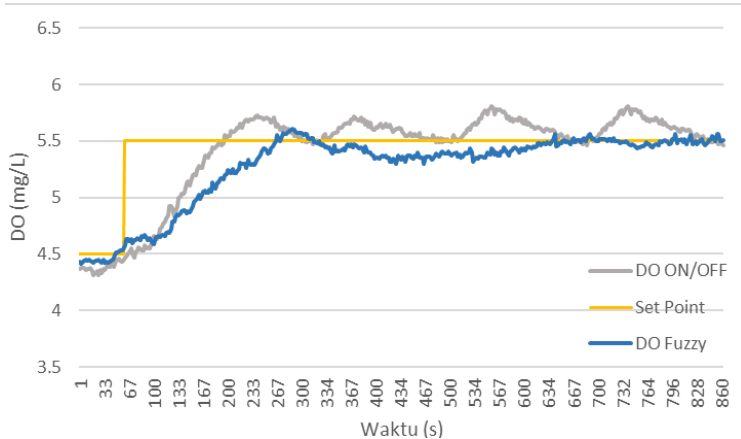
Gambar 4.7 Respon DO terhadap disturbance 30 L

Berdasarkan hasil uji, dapat dilihat bahwa DO yang turun sampai 4.5 mg/L kemudian naik sampai set point namun turun kembali. Penyebabnya hampir sama dengan percobaan dengan gangguan 20 L. Namun pada percobaan dengan 30 L gangguan, respon sedikit demi sedikit membaik pada menit ke 5 dan mulai berada di sekitar *setpoint*. Hal ini cukup cepat dengan gangguan yang diberikan begitu besar. *Error* DO berada pada rentang ± 0.17 mg/L.

4.4.5 Perbandingan Kontrol Fuzzy dan On/ Off

Untuk memastikan apakah sistem kontrol bekerja dengan baik, sistem kontrol dengan *fuzzy* dibandingkan dengan sistem kontrol dengan *on-off*. Hasil perbandingan performansi sistem kontrol bisa dilihat pada Gambar 4.8. Kontrol *on/off* menggunakan empat pompa aerator untuk menaikkan kadar oksigen. Dapat diamati bahwa kontrol *on/off* memiliki *rise time* yang lebih cepat yaitu 2,4 menit. Namun kontrol *on/off* selalu beresilasi diatas *set point* yang diinginkan dengan *error* sebesar 0.3 mg/L. Dibandingkan dengan

on/off, *fuzzy* walaupun memiliki *rise time* yang sedikit lebih lama yaitu 3.7 menit., namun stabil dan *ess* yang kecil yaitu dibawah 0.1 mg/L. Karena pada sistem yang sesungguhnya perubahan *set point* cukup jarang dilakukan dan menjaga kestabilan sistem kadar oksigen lebih penting, maka kontrol dengan logika *fuzzy* lebih tepat digunakan untuk mengendalikan kadar oksigen terlarut.



Gambar 4.8 Perbandingan respon DO dengan *fuzzy* dan *on/off* (4 pompa)

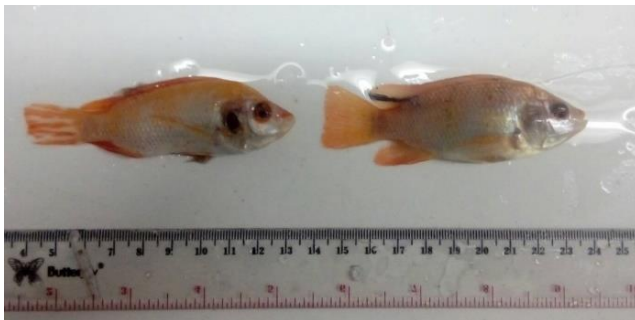
4.4. Laju Pertumbuhan Akuaponik

Untuk mengetahui pengaruh penggunaan sistem kontrol, dilakukan pengukuran terhadap pertumbuhan ikan dan tanaman. Pengukuran dilakukan selama selang waktu 10 hari dengan paramater pertambahan berat ikan dan dan pertambahan berat tanaman.

Tabel 4.3 Perbandingan Laju Pertumbuhan Ikan

Sample Ikan	Panjang Tubuh (cm)			
	Dikendalikan		Tidak Dikendalikan	
	Hari ke-1	Hari ke-10	Hari ke-1	Hari ke-10
1	6.8	7.6	7.1	7.6
2	7.1	7.6	7.2	7.6
3	7.1	7.6	7.0	7.4
4	7.4	8.1	7.2	7.6
5	7.0	7.7	7.2	7.7
6	7.2	8.0	7.2	7.6
7	6.8	7.3	7.4	7.8
8	6.9	7.7	6.8	7.3
9	7.0	7.6	7.4	7.9
10	7.4	8.1	7.3	7.8
Total (gr)	70.7	77.3	71.8	76.3
Perubahan (gr)		6.6		4.5
Rata-rata (gr)		0.66		0.45

Dari Tabel 4.2 didapatkan bahwa terjadi perbedaan rata-rata pertambahan berat ikan yaitu 6.6 gram untuk DO yang dikendalikan dan 4.5 gram untuk DO yang tidak dikendalikan. Terdapat kenaikan sebesar 2.1 gram atau meningkat 25% selama 10 hari.

**Gambar 4.9** Perbandingan Ikan dengan DO Dikendalikan (kanan) dan Tidak (kiri)

Selain penambahan berat juga terjadi perbedaan panjang dari tubuh ikan sendiri seperti terlihat pada Gambar 4.10. Ikan yang hidup pada lingkungan perairan dengan kadar oksigen yang cukup mampu melakukan metabolisme dengan baik. Sedangkan untuk ikan yang hidup dengan kadar oksigen terlarut yang rendah, metabolismenya akan terganggu dan nafsu makannya kurang. Pada ikan yang kadar DO nya tidak dikendalikan, terlihat luka hitam akibat terserang penyakit. Hal ini disebabkan oleh kadar oksigen yang rendah membuat air menjadi tempat pertumbuhan penyakit yang subur, karena penguraian racun oleh bakteri menjadi terhambat.

Dari Tabel 4.3 didapatkan bahwa terjadi perbedaan rata-rata pertumbuhan tinggi tanaman yaitu 2.3 cm untuk DO yang dikendalikan dan 1.1 cm untuk DO yang tidak dikendalikan. Terdapat kenaikan sebesar 1.2 cm.

Tabel 4.4 Perbandingan Laju Pertumbuhan Tanaman

Sample Tanaman	Tinggi Tanaman (cm)					
	Dikendalikan			Tidak Dikendalikan		
	Hari ke-1	Hari ke-10	Perubahan	Hari ke-1	Hari ke-10	Perubahan
1	9.90	11.43	1.53	9.99	11.28	1.29
2	10.11	12.15	2.04	10.26	11.19	0.93
3	9.87	12.24	2.37	9.84	11.04	1.20
4	10.20	12.93	2.73	9.87	10.80	0.93
5	10.01	12.675	2.67	10.23	11.28	1.05
Perubahan		Rata-rata	2.3	Perubahan		Rata-rata
		/hari	0.2268			/hari
						0.108

Pertambahan panjang tanaman bisa dilihat pada gambar 4.11. Terlihat bahwa untuk akuaponik dengan DO dikontrol pada 5.5 mg/L terjadi pertumbuhan tinggi yang lebih cepat. Hal ini disebabkan karena dengan nilai DO yang mencukupi membuat tanaman mampu menyerap nutrisi dengan baik sehingga pertumbuhannya lebih cepat.



Gambar 4.10 Perbandingan sayur dengan DO dikendalikan (kiri) dan tidak (kanan)

Halaman ini memang dikosongkan

BAB V

PENUTUP

5.1. Kesimpulan

- 1) Sistem kontrol kadar oksigen terlarut pada akuaponik menggunakan kontrol logika *fuzzy* telah dibuat dengan masukan *error* DO dan % saturasi dan keluaran jumlah pompa aerator untuk memanipulasi laju aerasi, berhasil mengendalikan kadar oksigen terlarut akuaponik pada *setpoint* 5.5 mg/L dengan performansi *maximum overshoot* (Mp) 1.27%, *rise time* (t_r) 211 detik, *setling time* (t_s) 574 detik, *ess* 0.07 mg/ L atau 1%.dan mengikuti perubahan *set point* yang diberikan dari 3 mg/L menuju 4,5 mg/L menuju 5.5 mg/L menuju 6.5 mg/L serta mampu mengatasi gangguan dengan baik berupa 10 L, 20L, dan 30 L air dengan kadar ± 1 mg/L.
- 2) Penggunaan sistem pengendali kadar oksigen terlarut dengan algoritma *fuzzy logic controller* pada akuaponik berhasil meningkatkan laju pertumbuhan ikan 0.21 gram dan peningkatan tinggi tanaman rata-rata 1.2 cm dalam jangka waktu 10 hari.

5.2. Saran

- 1) Dilakukan pemodelan DO pada akuaponik sebagai identifikasi sistem.
- 2) Pengamatan pengaruh penggunaan sistem kontrol dilakukan mulai benih hingga panen dan yang di ukur tidak hanya berat ikan atau tinggi tanaman saja.
- 3) Ditambahkan pengaruh umur ikan dan tanaman sebagai *input* dari *fuzzy logic controller* karena perbedaan umur mempengaruhi karakteristik DO akuaponik.
- 4) Digabungkan dengan sistem kontrol variabel lainnya seperti pH dan EC dan diteliti kaitannya satu variabel dengan yang lainnya.

Halaman ini memang dikosongkan

DAFTAR PUSTAKA

- Eutech Instruments. (1997). *Dissolved Oxygen Electrodes*. Tech Tips.
- Amand, L., Olsson, G., & Carlsson, B. (2013). Aeration control - a review. *Water Science and Technology*, 67(11), 2374-23398.
- Amri, K., & Khairuman. (2003). *Budidaya Ikan Nila Secara Intensif*. Jakarta: PT. Agromedia Pustaka.
- Atlas Scientific. (2016). *A practical guide for understanding dissolved oxygen reading*. New York: Atlas Scientific.
- Badan Standardisasi Nasional. (2009). *Produksi Ikan Nila (Oreochromis Niloticus Bleeker) Kelas Pembesaran di Kolam Air Tenang*. Jakarta: BSN.
- Cherif, M., Tirilly, Y., & Belanger, R. R. (1997). Effect of oxygen concentration on plant growth, lipidperoxidation, and receptivity of tomato roots to Pythium F under hydroponic conditions. *European Journal of Plant Pathology*, 103(3), 255-264.
- Finesse, LLC. (n.d.). *Dissolved Oxygen Sensor Primer*. Santa Clara: Finesse.
- Food and Agriculture Organization of the United Nations. (2011). *The State of the World's Land and Water Resources for Food and Agriculture*. London: Earthscan.
- Food and Agriculture Organization of United Nations. (2016). *The State of World Fisheries and Aquaculture*. Rome.
- Francis, R., & Floyd. (2014). *Dissolved Oxygen for Fish Production*. Gainesville: University of Florida.
- Gislerod, H. R., & Kempton, R. J. (1983, May). The oxygen content of flowing nutrient solutions used for cucumber and tomato culture. *Scientia Horticulturae*, 20, 23-33.
- Goddek, S., Delaide, B., Mankasingh, U., & Ragnarsdottir, K. V. (2015). Challenges of Sustainable and Commercial Aquaponics. *Sustainability*, 7, 4199-4224.

- Hargreaves, J. A., & Tucker, C. S. (2002). *Measuring Dissolved Oxygen Concentration In Aquaculture*. Stoneville: Southern Regional Aquaculture Center.
- Jegatheesan, V., Shu, L., & Visvanathan, C. (2011). Aquaculture Effluent: Impacts and Remedies for Protecting the Environment and Human Health. *Encyclopedia of Environmental Health*, 123-135.
- Ladon, L. (2001). *Electrochemistry*. Towson University.
- Larsen, D. (2014, 4 20). *Electrochemical Cell Conventions*. Diambil kembali dari UCDavis ChemWiki: https://chem.libretexts.org/Core/Analytical_Chemistry/Electrochemistry/Voltaic_Cells/Electrochemical_Cell_Conventions
- Mallya, Y. J. (2007). *The Effects of Dissolved Oxygen on Fish Growth in Aquaculture*. Reykjavik, Iceland: Ministry of Natural Resources and Tourism.
- McGraw, W., Teichert-Coddington, D. R., Rouse, D. B., & Boyd, C. E. (2001, August). Higher minimum dissolved oxygen concentrations increase penaeid shrimp yields in earthen ponds. *Aquaculture*, 199, 311-321.
- Mudjiman, A. (2001). *Makanan Ikan*. Jakarta: Penebar Swadaya.
- Ogata, K. (2010). *Modern Control Engineering 5th Edition*. New Jersey: Prentice Hall.
- Papoutsoglou, S. E., & Tziha, G. (1996, May). Blue tilapia (*Oreochromis aureus*) growth rate in relation to dissolved oxygen concentration under recirculated water conditions. *Aquacultural Engineering*, 15(3), 181-192.
- Piotrowski, R., & Skiba, A. (2015). Nonlinear Fuzzy Control System for Dissolved Oxygen with Aeration System in Sequencing Batch Reactor. *Information Technology and Control*, 44(2), 182-195.
- Rakocy, J., & Hargreaves, J. (1993). Integration of Vegetable Hydroponic with Fish Culture: A Review In. *Techniques for Modern Aquaculture*, 21-23.
- Santosa, B. (2001). *Budidaya Ikan Nila*. Yogyakarta: Kanisius.

- Sari, I. G. (2014). *Budidaya Ikan dan Tanaman dengan Sistem Akuaponik*. Purwokerto: Universitas Soedirman.
- Seginer, I., & Mozes, N. (2012). A note on oxygen supply in RAS: The effect of water temperature. *Aquaculture Engineering*, 50, 45-54.
- Soffer, H., Burger, D. W., & Lieth, J. H. (1991, January). Plant growth and development of Chrysanthemum and Ficus in aero-hydroponics: response to low dissolved oxygen concentrations. *Scientia Horticulturae*, 45(3-4), 287-294.
- Sucipto, A., & Prihartono, R. E. (2005). *Pembesaran Nila Merah Bangkok*. Jakarta: Penebar Swadaya.
- Thermo Electron Corporation. (n.d.). *Orion DO Theory*. In *The Technical Edge*. Thermo Electron Corporation.
- Timmons, M. B., Ebeling, J. M., Wheaton, F. W., Summerfelt, S. T., & Vinci, B. J. (2007). Recirculating aquaculture.
- Tran-Ngoc, K. T., Dinh, N. T., Nguyen, T. H., Roem, A. J., Schrama, J. W., & Varreth, J. A. (2016, September). Interaction between dissolved oxygen concentration and diet composition on growth, digestibility and intestinal health of Nile tilapia. *Aquaculture*, 462, 101-108.
- Watt, M. K. (2000). *A Hydrologic Primer for New Jersey Watershed Management (Water-Resources Investigation Report 00-4140)*. West Trenton: U.S. Geological Survey.
- Wilkin, R. T., & Ptacek, C. J. (2000). Field Measurement of Geochemical Redox Parameters. *Workshop on Monitoring Oxidation-Reduction Processes for Groundwater Restoration*.
- World Resources Institute. (2015, May). *The Great Balancing Act*. Diambil kembali dari World Resources Institute: <http://www.wri.org/publication/great-balancing-act>
- Yoshida, S., Kitano, M., & Eguchi, H. (1997). Growth of lettuce plants (*Lactuca sativa* L.) under control of dissolved O₂ concentration in hydroponics. *Biotronic*, 29, 39-45.
- YSI. (2009). *The Dissolved Oxygen Handbook*. YSI Incorporated.
- YSI Incorporated. (2008). *Pro20 User Manual*. YSI Incorporated.

LAMPIRAN A
HASIL PERANCANGAN MEKANIK DAN ELEKTRIK



Gambar C.1 Hasil Perancangan Akuaponik yang Digunakan sebagai Objek Percobaan Sistem Kontrol

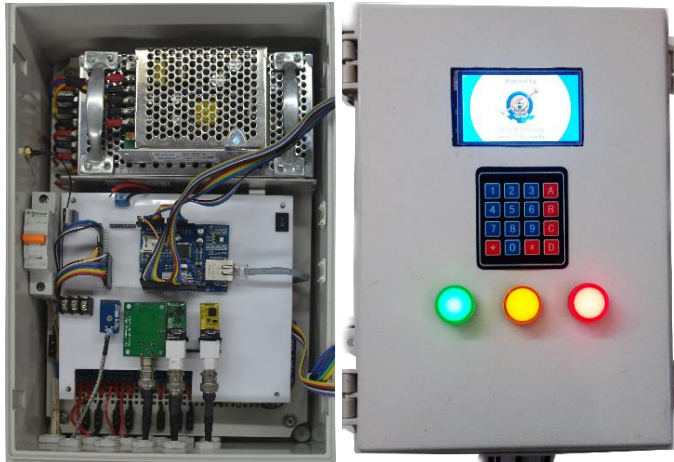


(a)

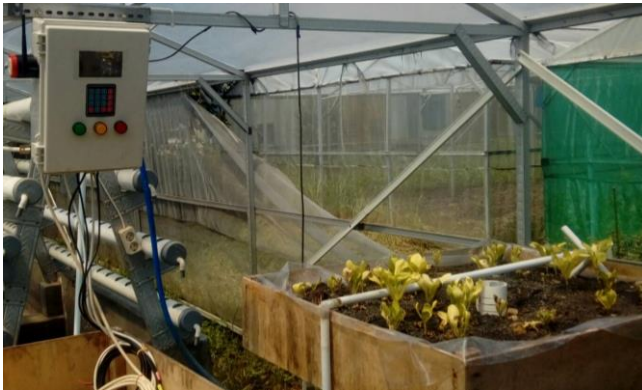


(b)

Gambar C.2 (a) Sensor DO pada Kolam Percobaan (b) Empat Pompa Aerator yang Digunakan untuk Mengatur Laju Aerasi



Gambar C.3 Hasil Perancangan Sistem Kontrol DO



Gambar C.4 Alat Terpasang pada Akuaponik

LAMPIRAN B

PROGRAM FUZZY PADA ARDUINO

```
#include "fis_header.h"

// Number of inputs to the fuzzy inference system
const int fis_gcI = 2;
// Number of outputs to the fuzzy inference system
const int fis_gcO = 1;
// Number of rules to the fuzzy inference system
const int fis_gcR = 9;

FIS_TYPE g_fisInput[fis_gcI];
FIS_TYPE g_fisOutput[fis_gcO];

// Setup routine runs once when you press reset:
void setup()
{
    // initialize the Analog pins for input.
    // Pin mode for Input: error_DO
    pinMode(0 , INPUT);
    // Pin mode for Input: Range
    pinMode(1 , INPUT);

    // initialize the Analog pins for output.
    // Pin mode for Output: aeration_pump
    pinMode(2 , OUTPUT);}

// Loop routine runs over and over again forever:
void loop()
{
    // Read Input: error_DO
```

```

g_fisInput[0] = analogRead(0);
// Read Input: Range
g_fisInput[1] = analogRead(1);

g_fisOutput[0] = 0;

fis_evaluate();
// Set output vlaue: aeration_pump
analogWrite(2 , g_fisOutput[0]);}

//*****
Support functions for Fuzzy Inference System
//*****
// Triangular Member Function
FIS_TYPE fis_trimf(FIS_TYPE x, FIS_TYPE* p)
{
    FIS_TYPE a = p[0], b = p[1], c = p[2];
    FIS_TYPE t1 = (x - a) / (b - a);
    FIS_TYPE t2 = (c - x) / (c - b);
    if ((a == b) && (b == c)) return (FIS_TYPE) (x == a);
    if (a == b) return (FIS_TYPE) (t2*(b <= x)*(x <= c));
    if (b == c) return (FIS_TYPE) (t1*(a <= x)*(x <= b));
    t1 = min(t1, t2);
    return (FIS_TYPE) max(t1, 0);}

// Trapezoidal Member Function
FIS_TYPE fis_trapmf(FIS_TYPE x, FIS_TYPE* p)
{
    FIS_TYPE a = p[0], b = p[1], c = p[2], d = p[3];
    FIS_TYPE t1 = ((x <= c) ? 1 : ((d < x) ? 0 : ((c != d) ? ((d - x) /
(d - c)) : 0)));

```

```

    FIS_TYPE t2 = ((b <= x) ? 1 : ((x < a) ? 0 : ((a != b) ? ((x - a) /
(b - a)) : 0)));
    return (FIS_TYPE) min(t1, t2);}

```

```

FIS_TYPE fis_min(FIS_TYPE a, FIS_TYPE b){
    return min(a, b);}

```

```

FIS_TYPE fis_max(FIS_TYPE a, FIS_TYPE b){
    return max(a, b);}

```

```

FIS_TYPE fis_array_operation(FIS_TYPE *array, int size,
_FIS_ARR_OP pfnOp)
{
    int i;
    FIS_TYPE ret = 0;
    if (size == 0) return ret;
    if (size == 1) return array[0];
    ret = array[0];
    for (i = 1; i < size; i++)
    {
        ret = (*pfnOp)(ret, array[i]);}
    return ret;}

```

```

//*****

```

Data for Fuzzy Inference System

```

//*****

```

```

// Pointers to the implementations of member functions

```

```

_FIS_MF fis_gMF[] =
{
    fis_trimf, fis_trapmf};

```

```

// Count of member function for each Input

```

```
int fis_gIMFCount[] = { 5, 3 };
```

```
// Count of member function for each Output
```

```
int fis_gOMFCount[] = { 5 };
```

```
// Coefficients for the Input Member Functions
```

```
FIS_TYPE fis_gMFI0Coeff1[] = { -0.4, 0, 0.25 };
```

```
FIS_TYPE fis_gMFI0Coeff2[] = { 0.25, 0.5, 0.75 };
```

```
FIS_TYPE fis_gMFI0Coeff3[] = { 0.75, 1, 1.4 };
```

```
FIS_TYPE fis_gMFI0Coeff4[] = { 0, 0.25, 0.5 };
```

```
FIS_TYPE fis_gMFI0Coeff5[] = { 0.5, 0.75, 1 };
```

```
FIS_TYPE* fis_gMFI0Coeff[] = { fis_gMFI0Coeff1,  
fis_gMFI0Coeff2, fis_gMFI0Coeff3, fis_gMFI0Coeff4,  
fis_gMFI0Coeff5 };
```

```
FIS_TYPE fis_gMFI1Coeff1[] = { -45, -5, 10, 25 };
```

```
FIS_TYPE fis_gMFI1Coeff2[] = { 10, 20, 45, 55 };
```

```
FIS_TYPE fis_gMFI1Coeff3[] = { 45, 55, 105, 145 };
```

```
FIS_TYPE* fis_gMFI1Coeff[] = { fis_gMFI1Coeff1,  
fis_gMFI1Coeff2, fis_gMFI1Coeff3 };
```

```
FIS_TYPE** fis_gMFICoeff[] = { fis_gMFI0Coeff,  
fis_gMFI1Coeff };
```

```
// Coefficients for the Input Member Functions
```

```
FIS_TYPE fis_gMFO0Coeff1[] = { -1, 0, 1 };
```

```
FIS_TYPE fis_gMFO0Coeff2[] = { 0, 1, 2 };
```

```
FIS_TYPE fis_gMFO0Coeff3[] = { 1, 2, 3 };
```

```
FIS_TYPE fis_gMFO0Coeff4[] = { 2, 3, 4 };
```

```
FIS_TYPE fis_gMFO0Coeff5[] = { 3, 4, 5 };
```

```
FIS_TYPE* fis_gMFO0Coeff[] = { fis_gMFO0Coeff1,  
fis_gMFO0Coeff2, fis_gMFO0Coeff3, fis_gMFO0Coeff4,  
fis_gMFO0Coeff5 };
```

```
FIS_TYPE** fis_gMFOCoeff[] = { fis_gMFO0Coeff };
```

```

// Input membership function set
int fis_gMFI0[] = { 0, 0, 0, 0, 0 };
int fis_gMFI1[] = { 1, 1, 1 };
int* fis_gMFI[] = { fis_gMFI0, fis_gMFI1 };

// Output membership function set
int fis_gMFO0[] = { 0, 0, 0, 0, 0 };
int* fis_gMFO[] = { fis_gMFO0 };

// Rule Weights
FIS_TYPE fis_gRWeight[] = { 1, 1, 1, 1, 1, 1, 1, 1, 1 };

// Rule Type
int fis_gRType[] = { 1, 1, 1, 1, 1, 1, 1, 1, 1 };

// Rule Inputs
int fis_gRI0[] = { 1, 0 };
int fis_gRI1[] = { 4, 3 };
int fis_gRI2[] = { 4, 1 };
int fis_gRI3[] = { 5, 3 };
int fis_gRI4[] = { 5, 1 };
int fis_gRI5[] = { 3, 0 };
int fis_gRI6[] = { 2, 2 };
int fis_gRI7[] = { 2, 1 };
int fis_gRI8[] = { 2, 3 };
int* fis_gRI[] = { fis_gRI0, fis_gRI1, fis_gRI2, fis_gRI3, fis_gRI4,
fis_gRI5, fis_gRI6, fis_gRI7, fis_gRI8 };

// Rule Outputs
int fis_gRO0[] = { 2 };
int fis_gRO1[] = { 2 };
int fis_gRO2[] = { 3 };

```

```

int fis_gRO3[] = { 4 };
int fis_gRO4[] = { 5 };
int fis_gRO5[] = { 5 };
int fis_gRO6[] = { 3 };
int fis_gRO7[] = { 4 };
int fis_gRO8[] = { 2 };
int* fis_gRO[] = { fis_gRO0, fis_gRO1, fis_gRO2, fis_gRO3,
fis_gRO4, fis_gRO5, fis_gRO6, fis_gRO7, fis_gRO8 };

// Input range Min
FIS_TYPE fis_gIMin[] = { 0, };

// Input range Max
FIS_TYPE fis_gIMax[] = { 1, };

// Output range Min
FIS_TYPE fis_gOMin[] = { 0 };

// Output range Max
FIS_TYPE fis_gOMax[] = { 4 };

//*****//
    Data dependent support functions for Fuzzy Inference System
//*****//
FIS_TYPE fis_MF_out(FIS_TYPE** fuzzyRuleSet, FIS_TYPE x,
int o)
{
    FIS_TYPE mfOut;
    int r;

    for (r = 0; r < fis_gcR; ++r)
    {

```



```

int index = fis_gRO[r][o];
if (index > 0){
    index = index - 1;
    mfOut = (fis_gMF[fis_gMFO[o][index]])(x,
        fis_gMFOCoeff[o][index]);}
else if (index < 0){
    index = -index - 1;
    mfOut = 1 - (fis_gMF[fis_gMFO[o][index]])(x,
        fis_gMFOCoeff[o][index]);}
else{
    mfOut = 0;}

fuzzyRuleSet[0][r] = fis_min(mfOut, fuzzyRuleSet[1][r]);}
return fis_array_operation(fuzzyRuleSet[0], fis_gcR, fis_max);}

FIS_TYPE fis_defuzz_centroid(FIS_TYPE** fuzzyRuleSet, int o)
{
    FIS_TYPE step = (fis_gOMax[o] - fis_gOMin[o]) /
(FIS_RESOLUTION - 1);
    FIS_TYPE area = 0;
    FIS_TYPE momentum = 0;
    FIS_TYPE dist, slice;
    int i;

    // calculate the area under the curve formed by the MF outputs
    for (i = 0; i < FIS_RESOLUTION; ++i){
        dist = fis_gOMin[o] + (step * i);
        slice = step * fis_MF_out(fuzzyRuleSet, dist, o);
        area += slice;
        momentum += slice*dist;
    }
}

```

```

    return ((area == 0) ? ((fis_gOMax[o] + fis_gOMin[o]) / 2) :
(momentum / area));}

```

```

//*****

```

Fuzzy Inference System

```

//*****

```

```

void fis_evaluate()

```

```

{

```

```

    FIS_TYPE fuzzyInput0[] = { 0, 0, 0, 0, 0 };

```

```

    FIS_TYPE fuzzyInput1[] = { 0, 0, 0 };

```

```

    FIS_TYPE* fuzzyInput[fis_gcI] = { fuzzyInput0, fuzzyInput1, };

```

```

    FIS_TYPE fuzzyOutput0[] = { 0, 0, 0, 0, 0 };

```

```

    FIS_TYPE* fuzzyOutput[fis_gcO] = { fuzzyOutput0, };

```

```

    FIS_TYPE fuzzyRules[fis_gcR] = { 0 };

```

```

    FIS_TYPE fuzzyFires[fis_gcR] = { 0 };

```

```

    FIS_TYPE* fuzzyRuleSet[] = { fuzzyRules, fuzzyFires };

```

```

    FIS_TYPE sW = 0;

```

```

// Transforming input to fuzzy Input

```

```

int i, j, r, o;

```

```

for (i = 0; i < fis_gcI; ++i){

```

```

    for (j = 0; j < fis_gIMFCount[i]; ++j){

```

```

        fuzzyInput[i][j] =

```

```

        (fis_gMF[fis_gMFI[i][j]])(g_fisInput[i],

```

```

        fis_gMFICoeff[i][j]);

```

```

    }}

```

```

int index = 0;

```

```

for (r = 0; r < fis_gcR; ++r){

```

```

    if (fis_gRType[r] == 1){

```

```

        fuzzyFires[r] = FIS_MAX;

```

```

        for (i = 0; i < fis_gcI; ++i){

```

```

    index = fis_gRI[r][i];
    if (index > 0)
        fuzzyFires[r] = fis_min(fuzzyFires[r],
            fuzzyInput[i][index - 1]);
    else if (index < 0)
        fuzzyFires[r] = fis_min(fuzzyFires[r], 1 -
            fuzzyInput[i][-index - 1]);
    else
        fuzzyFires[r] = fis_min(fuzzyFires[r], 1);} }
else{
    fuzzyFires[r] = FIS_MIN;
    for (i = 0; i < fis_gcI; ++i){
        index = fis_gRI[r][i];
        if (index > 0)
            fuzzyFires[r] = fis_max(fuzzyFires[r],
                fuzzyInput[i][index - 1]);
        else if (index < 0)
            fuzzyFires[r] = fis_max(fuzzyFires[r], 1 -
                fuzzyInput[i][-index - 1]);
        else
            fuzzyFires[r] = fis_max(fuzzyFires[r], 0);} }

    fuzzyFires[r] = fis_gRWeight[r] * fuzzyFires[r];
    sW += fuzzyFires[r];}

if (sW == 0){
    for (o = 0; o < fis_gcO; ++o){
        g_fisOutput[o] = ((fis_gOMax[o] + fis_gOMin[o]) / 2);} }
else{
    for (o = 0; o < fis_gcO; ++o){
        g_fisOutput[o] = fis_defuzz_centroid(fuzzyRuleSet, o);} }
}

```


LAMPIRAN C

SOURCE CODE FIRMWARE

```
unsigned long time;
long lastSDLog = 0;
float SDinterval = 2000;
long lastThingerLog = 0;
float thingerInterval = 10000;

//----- ETHERNET & PLATFORM -----//
#include <Ethernet.h>
#include <ThingerEthernet.h>
#define USERNAME "alberto_rc"
#define DEVICE_ID "mega2560_1"
#define DEVICE_CREDENTIAL "quatic_4.1"

ThingerEthernet thing(USERNAME, DEVICE_ID,
DEVICE_CREDENTIAL);
EthernetClient client;
byte mac[] = {0x00, 0xAA, 0xBB, 0xCC, 0xDE, 0x02};

//----- DISPLAY & SD -----//
#include <SPI.h>
#define USE_SDFAT
#include <SdFat.h>
SdFat SD;
String dataLog = "";

#include <Adafruit_GFX.h>
#include <Adafruit_TFTLCD.h>
#include <MCUFRIEND_kbv.h>
MCUFRIEND_kbv tft;
#include <Fonts/FreeMonoBold12pt7b.h>
```

```

#include <Fonts/FreeSans9pt7b.h>
#define SD_CS A6
File root;
//----- TEMPERATURE -----//
#include "max6675.h"
int thermoDO = 23;
int thermoCS = 25;
int thermoCLK = 27;
float tempRead = 0;
float tempOut;
MAX6675 thermocouple(thermoCLK, thermoCS, thermoDO);
//----- EC&DO -----//
String inputSensor = "";
String sensorDO = "";
boolean inputSensorComplete = false;
boolean sensorDOcomplete = false;
char sensorDOarray[30];
char *DO;
char *SAT;
float f_DO;
float f_SAT;
float setPointDO = 6;
float DOtempComp = 30;
float errorDOTolerance = 1;
float errorDO;
//----- OUTPUT -----//
#define aerator1      22
#define aerator2      24
#define aerator3      26
#define aerator4      28
#define indicatorRed   47
#define indicatorYellow 49

```

```

#define indicatorGreen 42
#define alarm          40
bool aer1state;
bool aer2state;
bool aer3state;
bool aer4state;
bool DOcalibrate = 0;
float controlMode = 1; //0 = AUTO, 1 = MANUAL
float OnOffMode = 4;

//===== SETUP =====//
void setup() {
  pinMode(aerator1, OUTPUT);      digitalWrite(aerator1, HIGH);
  pinMode(aerator2, OUTPUT);      digitalWrite(aerator2, HIGH);
  pinMode(aerator3, OUTPUT);      digitalWrite(aerator3, HIGH);
  pinMode(aerator4, OUTPUT);      digitalWrite(aerator4, HIGH);
  pinMode(indicatorRed, OUTPUT);   digitalWrite(indicatorRed, HIGH);
  pinMode(indicatorGreen, OUTPUT); digitalWrite(indicatorGreen, HIGH);
  pinMode(indicatorYellow, OUTPUT); digitalWrite(indicatorYellow, HIGH);
  pinMode(alarm, OUTPUT);          digitalWrite(alarm, HIGH);

//----- THINGER IN -----//
  thing["Temperature"] >> [](pson& out){
    out = tempOut;};
  thing["DO"] >> [](pson& out){
    out = f_DO;};
  thing["%Saturation"] >> [](pson& out){
    out = f_SAT;};
  thing["Indikator Merah"] >> [](pson& out){
    out = (bool) digitalRead(indicatorRed);};
  thing["Indikator Kuning"] >> [](pson& out){
    out = (bool) digitalRead(indicatorYellow);};

```

```
thing["Indikator Hijau"] >> [](pson& out){
    out = (bool) digitalRead(indicatorGreen);};
```

```
//----- THINGER OUT -----//
```

```
thing["Mode"] << [](pson& in){
    if(in.is_empty()){
        in = controlMode;}
    else{
        controlMode = in;}
    tft.setTextSize(3);
    if (controlMode == 1){
        tft.setTextColor(GREEN, WHITE);
        tft.setCursor(360, 80);
        tft.print("AUTO ");}
    else if (controlMode == 2){
        tft.setTextColor(BLUE, WHITE);
        tft.setCursor(360, 80);
        tft.print("ON/OFF ");}
    else if (controlMode == 3){
        tft.setTextColor(BLACK, WHITE);
        tft.setCursor(360, 80);
        tft.print("MANUAL");}
};
```

```
thing["OnOff Mode"] << [](pson& in){
    if(in.is_empty()){
        in = OnOffMode;}
    else{
        OnOffMode = in;}};
```

```
thing["Aerator 1"] << [](pson& in){
    if(in.is_empty()){
```



```
in = (bool)digitalRead(aerator1);}
else{
  if (controlMode == 3){
    digitalWrite(aerator1, in ? HIGH : LOW);}
aer1print();};
```

```
thing["Aerator 2"] << [](pson& in){
if(in.is_empty()){
  in = (bool)digitalRead(aerator2);}
else{
  if (controlMode == 3){
    digitalWrite(aerator2, in ? HIGH : LOW);}
aer2print();};
```

```
thing["Aerator 3"] << [](pson& in){
if(in.is_empty()){
  in = (bool)digitalRead(aerator3);}
else{
  if (controlMode == 3){
    digitalWrite(aerator3, in ? HIGH : LOW);}
aer3print(); };
```

```
thing["Aerator 4"] << [](pson& in){
if(in.is_empty()){
  in = (bool)digitalRead(aerator4);}
else{
  if (controlMode == 3){
    digitalWrite(aerator4, in ? HIGH : LOW);}
aer4print();};
```

```
thing["Alarm"] << [](pson& in){
if(in.is_empty()){
```

```

    in = (bool)digitalRead(alarm);}
else{
    digitalWrite(alarm, in ? HIGH : LOW);});

thing["Set point DO"] << [](pson& in){
    if(in.is_empty()){
        in = setPointDO;}
    else{
        setPointDO = in;}};

thing["DO Temp. Compesation"] << [](pson& in){
    if(in.is_empty()){
        in = DOtempComp;}
    else{
        DOtempComp = in;
        String Tcompensate = "T," + String(DOtempComp) + "\r";
        Serial3.print(Tcompensate);});

thing["DO Calibration"] << [](pson& in){
    DOcalibrate = in;
    tft.setTextSize(3);
    if (DOcalibrate == HIGH){
        String DOcalibrate = "Cal\r";
        Serial3.print(DOcalibrate);});

thing["Datalog Interval"] << [](pson& in){
    if(in.is_empty()){
        in = SDinterval;}
    else{
        SDinterval = in;}};

thing["Thingerlog Interval"] << [](pson& in){

```

```
if(in.is_empty()){
  in = thingerInterval;}
else{
  thingerInterval = in; } };
```

```
Serial.begin(9600);
Serial3.begin(9600);
inputSensor.reserve(10);
sensorDO.reserve(30);
```

```
uint16_t ID = tft.readID();
tft.begin(ID);
tft.fillScreen(WHITE);
if (tft.height() > tft.width()) tft.setRotation(3);
tft.setTextColor(0xFFFF, 0x0000);
bool good = SD.begin(SD_CS);
root = SD.open(namebuf);
```

```
//----- HOMESCREEN -----//
tft.fillRect(110, 235, 280, 60, WHITE);
delay(300);
showBMP("roxtrade.bmp", 120, 75);
delay(1000);
tft.fillScreen(WHITE);
tft.setFont(&FreeMonoBold12pt7b);
tft.setTextColor(BLUE);
tft.setTextSize(2);
tft.setCursor(75, 110);
tft.print("Initializing");
tft.setCursor(50, 110);
tft.print("Try to connect");
```

```
while (Ethernet.begin(mac) == 0) {
  tft.setTextColor(RED, WHITE);
  tft.setCursor(150, 170);
  tft.print("Failed!");
}
tft.fillRect(100, 135, 300, 50, WHITE);
printIPAddress();
delay(2000);
tft.fillScreen(WHITE);

tft.fillRect(272, 28, 3, 264, GREEN);
tft.setFont(&FreeSans9pt7b);
tft.setTextColor(BLUE);
tft.setTextSize(2);
tft.setCursor(22,58);
tft.print("Water Quality");
tft.setCursor(323,58);
tft.print("Status");
tft.fillRect(10, 69, 247, 2, GREEN);
tft.fillRect(310, 69, 128, 2, GREEN);

tft.setFont();
tft.setTextColor(BLUE);
tft.setTextSize(2);
tft.setCursor(26, 83);
tft.print("DO :      mg/L");
tft.setCursor(26, 114);
tft.print("% Sat.:    %");
tft.setCursor(26, 145);
tft.print("Temp.:      C");
tft.setCursor(26, 176);
tft.print("SP DO:      mg/L");
```

```
tft.setCursor(26, 207);
tft.print("Error:      mg/L");
tft.setCursor(26, 238);
tft.print("%Stat:      ");
tft.setCursor(26, 269);
tft.print("Delta:      ");
tft.setCursor(290, 83);
tft.print("Mode: ");
tft.setCursor(290, 176);
tft.print("Aerator 1:");
tft.setCursor(290, 207);
tft.print("Aerator 2:");
tft.setCursor(290, 238);
tft.print("Aerator 3:");
tft.setCursor(290, 269);
tft.print("Aerator 4:");}
```

```
//===== LOOP de LOOP =====//
```

```
void loop() {
  if (inputSensorComplete == true) {
    Serial3.print(inputSensor);
    Serial3.print("\r");
    inputSensor = "";
    inputSensorComplete = false;}
}
```

```
if (sensorDOcomplete == true){
  showData();
  sensorDO = "";
  sensorDOcomplete = false; }
```

```
time = millis();
if (time - lastSDLog > SDinterval){
```

```

lastSDLog = time;
File dataFile = SD.open("datalog.txt", FILE_WRITE);
if (dataFile) {
    dataFile.println(dataLog);
    dataFile.close();}

thing.handle();
if (time - lastThingerLog > thingerInterval){
    lastThingerLog = time;
    thing.stream(thing["Temperature"]);
    thing.stream(thing["DO"]);
    thing.stream(thing["% Saturation"]);
    thing.stream(thing["Indikator Merah"]);
    thing.stream(thing["Indikator Kuning"]);
    thing.stream(thing["Indikator Hijau"]);}
if (controlMode == 1 || controlMode == 2){
    controlAction();
}

//----- PRINT IP -----//
void printIPAddress()
{
    tft.setTextColor(GREEN, WHITE);
    tft.setCursor(110, 165);
    tft.print("Connected!");
    tft.setCursor(125, 210);
    tft.setTextSize(1.5);
    tft.setTextColor(BLACK);
    tft.print("IP: ");
    for (byte thisByte = 0; thisByte < 4; thisByte++) {
        tft.print(Ethernet.localIP()[thisByte], DEC);
        tft.print(".");
    }
}

```

```
}}
```

```
//----- DATA READING & DISPLAY -----//
```

```
void showData(void) {  
  for (index; index < 72; index++){  
    phReading[index]= analogRead(phPin);  
    phTotal += phReading[index];  
    if (index==0 || index==18 || index==36 || index==54){  
      digitalWrite(thermoCS, LOW);  
      tempRead += thermocouple.readCelsius()-2.5; }  
    digitalWrite(thermoCS,HIGH);
```

```
  
    if (index == 72){  
      tempOut = tempRead/4;  
      tempRead = 0;  
      index = 0; }  
  

```

```
sensorDO.toCharArray(sensorDOarray, 30);  
DO = strtok(sensorDOarray, ",");  
SAT = strtok(NULL, ",");  
f_DO = atof(DO);  
f_SAT = atof(SAT);  
  

```

```
dataLog = String(time) + "," + String(tempOut) + "," + sensorDO
```

```
  
tft.setTextColor(BLACK, WHITE);  
tft.setTextSize(3);  
tft.fillRect(100, 78, 100, 25, WHITE);  
tft.setCursor(103, 78);  
tft.print(f_DO);  
tft.fillRect(100, 109, 100, 25, WHITE);  
tft.setCursor(103, 109);  
  

```

```

tft.print(f_SAT, 1);
tft.fillRect(100, 140, 100, 25, WHITE);
tft.setCursor(103, 140);
tft.print(tempOut, 1);
tft.fillRect(100, 171, 100, 25, WHITE);
tft.setCursor(103, 171);
tft.print(setPointDO);
tft.fillRect(100, 202, 100, 25, WHITE);
tft.setCursor(103, 202);
tft.print(errorDO);
if (f_SAT > 85){
    tft.fillRect(100, 233, 100, 25, WHITE);
    tft.setCursor(103, 233);
    tft.setTextColor(RED);
    tft.print("HIGH");}
if (f_SAT > 70 && f_SAT < 85){
    tft.fillRect(100, 233, 100, 25, WHITE);
    tft.setCursor(103, 233);
    tft.setTextColor(BLUE);
    tft.print("MID");}
if (f_SAT < 70){
    tft.fillRect(100, 233, 100, 25, WHITE);
    tft.setCursor(103, 233);
    tft.print("LOW");}

```

//----- SHOW STATUS -----//

```

void aer1print (void){
    aer1state = (bool)digitalRead(aerator1);
    tft.setTextSize(3);
    if (aer1state == 1){
        tft.setTextColor(WHITE, RED);
        tft.setCursor(410, 170);
    }
}

```



```
tft.print("OFF");}
else {
    tft.setTextColor(WHITE, BLUE);
    tft.setCursor(410, 170);
    tft.print("ON ");}

void aer2print (void){
    aer2state = (bool)digitalRead(aerator2);
    if (aer2state == HIGH){
        tft.setTextColor(WHITE, RED);
        tft.setCursor(410, 201);
        tft.print("OFF");}
    else {
        tft.setTextColor(WHITE, BLUE);
        tft.setCursor(410, 201);
        tft.print("ON ");}

void aer3print (void){
    aer3state = (bool)digitalRead(aerator3);
    if (aer3state == 1){
        tft.setTextColor(WHITE, RED);
        tft.setCursor(410, 233);
        tft.print("OFF");}
    else {
        tft.setTextColor(WHITE, BLUE);
        tft.setCursor(410, 233);
        tft.print("ON ");}}

void aer4print (void){
    aer4state = (bool)digitalRead(aerator4);
    if (aer4state == 1){
        tft.setTextColor(WHITE, RED);
```

```

    tft.setCursor(410, 264);
    tft.print("OFF");}
else {
    tft.setTextColor(WHITE, BLUE);
    tft.setCursor(410, 264);
    tft.print("ON ");}

//----- CONTROL ACTION -----//
void controlAction(void) {
    errorDO = setPointDO - f_DO;

    if (errorDO > errorDOTolerance){
        digitalWrite(indicatorYellow, LOW);}
    else {
        digitalWrite(indicatorYellow, HIGH);}

//----- ON/OFF -----//
    if (controlMode == 2){
        if (errorDO > 0){
            if (OnOffMode == 1){
                digitalWrite(aerator1, LOW);
                digitalWrite(aerator2, HIGH);
                digitalWrite(aerator3, HIGH);
                digitalWrite(aerator4, HIGH);}
            else if (OnOffMode == 2){
                digitalWrite(aerator1, LOW);
                digitalWrite(aerator2, LOW);
                digitalWrite(aerator3, HIGH);
                digitalWrite(aerator4, HIGH);}
            else if (OnOffMode == 3){
                digitalWrite(aerator1, LOW);
                digitalWrite(aerator2, LOW);

```

```

        digitalWrite(aerator3, LOW);
        digitalWrite(aerator4, HIGH);}
else if (OnOffMode == 4){
    digitalWrite(aerator1, LOW);
    digitalWrite(aerator2, LOW);
    digitalWrite(aerator3, LOW);
    digitalWrite(aerator4, LOW);} }
else {
    digitalWrite(aerator1, HIGH);
    digitalWrite(aerator2, HIGH);
    digitalWrite(aerator3, HIGH);
    digitalWrite(aerator4, HIGH);} }
aer1print();
aer2print();
aer3print();
aer4print();}

void serialEvent() {
    inputSensor = Serial.readStringUntil(13);
    inputSensorComplete = true;}

void serialEvent3() {
    sensorDO = Serial3.readStringUntil(13);
    sensorDOcomplete = true;}

```


BIODATA PENULIS



Nama lengkap penulis adalah Alberto Riolly Cahyantara, lahir di Gunungkidul tanggal 24 April 1995. Penulis telah menyelesaikan pendidikan formal, yaitu SDN Grogol IV, SMPN 1 Wonosari, SMAN 1 Wonosari. Setelah lulus dari SMA, penulis diterima di Departemen Teknik Fisika ITS. Selama kuliah, penulis telah aktif dalam beberapa organisasi dan kepanitiaan, seperti menjadi Kepala Departemen Kewirausahaan di HMTF ITS periode 2014-2015 dan Kepala Divisi *Maintenenace* Lab. Rekayasa Instrumentasi dan Kontrol 2015-2016. Penulis memiliki ketertarikan dalam penelitian berkaitan dengan teknologi tepat guna dibidang pertanian, perkebunan, dan perikanan. Lingkup teknologi yang dikuasai penulis meliputi instrumentasi dan kontrol, otomasi, optimasi, dan *artificial intelligence*. Bagi pembaca yang memiliki kritik, saran atau ingin berdiskusi seputar topik tersebut, dapat menghubungi penulis melalui *email*: albert.r.cahyantara@gmail.com atau menghubungi di nomor berikut 081229773422.

